

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

DIPLOMOVÁ PRÁCE

Brno, 2018

Bc. Juraj Gierť



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

**FAKULTA ELEKTROTECHNIKY
A KOMUNIKAČNÍCH TECHNOLOGIÍ**

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV TELEKOMUNIKACÍ

DEPARTMENT OF TELECOMMUNICATIONS

**EMULACE INFRASTRUKTURNÍ JEDNOTKY PRO SYSTÉM
INTELIGENTNÍ DOPRAVY**

EMULATION OF INFRASTRUCTURE UNIT FOR INTELLIGENT TRANSPORT SYSTEM

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. Juraj Giertl

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Miroslav Balík, Ph.D.

BRNO 2018

Diplomová práce

magisterský navazující studijní obor **Telekomunikační a informační technika**

Ústav telekomunikací

Student: Bc. Juraj Giertl

ID: 164271

Ročník: 2

Akademický rok: 2017/18

NÁZEV TÉMATU:

Emulace infrastrukturní jednotky pro systém inteligentní dopravy

POKYNY PRO VYPRACOVÁNÍ:

Nastudujte a popište technologii pro bezdrátovou komunikaci typu vozidlo-vozdlo a vozidlo-infrastruktura na bázi ETSI ITS-G5 z hlediska standardizace i probíhajících testovacích provozů v EU. Srovnajte ji se standardy IEEE 1609.x DSRC/WAVE a situací v USA. V detailu se zaměřte na světelné křižovatky, výstražné značení a popište protokol výměny příslušných zpráv SPaT, MAP a RSA dle existujících standardů. Implementujte aplikaci pro emulaci infrastrukturního bodu na vývojovém kitu firmy ALPS Electric obsahujícím V2X rádiový modul a vestavěný systém s Linux OS. Ověřte na aplikační úrovni funkčnost systému při testu s druhým kitem s testovací aplikací reprezentující vozidlo.

DOPORUČENÁ LITERATURA:

[1] Intelligent Transport Systems; Communications Architecture, European Standard (Telecommunications series) ETSI EN 302 665 v1.1.1, 2010

[2] Dedicated Short Range Communications (DSRC) Message Set Dictionary. Surface Vehicle Standard J2735, SAE International, 2016

Termín zadání: 5.2.2018

Termín odevzdání: 21.5.2018

Vedoucí práce: Ing. Miroslav Balík, Ph.D.

Konzultant: Ing. Josef Nevrlý, ALPS, Ing. Václav Mecerod, ALPS

prof. Ing. Jiří Mišurec, CSc.
předseda oborové rady

UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

ABSTRAKT

Práca sa zaoberá štúdiom komunikačných modelov pre kooperatívne systémy inteligentnej dopravy a vývojom testovacej aplikácie. Vybrané komunikačné modely ETSI ITS-G5 a IEEE 1609.x DSRC/WAVE sú popísané k referenčnému modelu ISO/OSI. Pre každý model sú opísané princípy komunikácie, ktoré systémy inteligentnej dopravy prinášajú. Následne sú podrobne rozpísané štruktúry a účel správ pre prenos informácií o hazardných situáciách, geometrií križovatky, fázy a stavy svetelnej signalizácie križovatky. Na základe týchto štruktúr a ďalších požiadaviek je vytvorená aplikácia, ktorá umožňuje jednoduché definovanie výstražných správ, geometrie križovatiek a jej stavov svetelnej signalizácie.

KLÚČOVÉ SLOVÁ

DSRC, ETSI ITS-G5, ITS, MAP, RSA, RSU, SPaT, V2X, V2I, V2V, WAVE

ABSTRACT

The thesis deals with the study of communication models for cooperative intelligent transport systems and the development of the application used for testing. The ETSI ITS-G5 and IEEE 1609.x DSRC/WAVE communication stacks were compared to standardizes layered ISO/OSI reference model. The basic principles of communication in intelligent transport systems are described for each model. Besides that the common messages structures for defining alert messages, the intersection geometry and traffic lights signals are described in further detail. Based on these structures and other requirements, an application is created that allows easy definition of alert messages, intersection geometry and its traffic light states.

KEYWORDS

DSRC, ETSI ITS-G5, ITS, MAP, RSA, RSU, SPaT, V2X, V2I, V2V, WAVE

GIERTL, Juraj. *Emulace infrastrukturní jednotky pro systém inteligentní dopravy*. Brno, Rok, 76 s. Diplomová práca. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací. Vedúci práce: Ing. Miroslav Balík, Ph.D.

VYHLÁSENIE

Vyhlasujem, že som svoju diplomovú prácu na tému „Emulace infrastrukturní jednotky pro systém inteligentní dopravy“ vypracoval samostatne pod vedením vedúceho diplomovej práce, využitím odbornej literatúry a ďalších informačných zdrojov, ktoré sú všetky citované v práci a uvedené v zozname literatúry na konci práce.

Ako autor uvedenej diplomovej práce ďalej vyhlasujem, že v súvislosti s vytvorením tejto diplomovej práce som neporušil autorské práva tretích osôb, najmä som nezasiahol nedovoleným spôsobom do cudzích autorských práv osobnostných a/alebo majetkových a som si plne vedomý následkov porušenia ustanovenia § 11 a nasledujúcich autorského zákona Českej republiky č. 121/2000 Sb., o práve autorskom, o právach súvisiacich s právom autorským a o zmene niektorých zákonov (autorský zákon), v znení neskorších predpisov, vrátane možných trestnoprávnych dôsledkov vyplývajúcich z ustanovenia časti druhej, hlavy VI. diel 4 Trestného zákoníka Českej republiky č. 40/2009 Sb.

Brno

.....

podpis autora

POĎAKOVANIE

Rád by som poďakoval svojim konzultantom Ing. Josefovi Nevrlému a Ing. Václavovi Mecerodovi, za odborné vedenie, za pomoc a rady pri spracovaní tejto práce a firme ALPS Electric Czech s.r.o. za technickú podporu a poskytnuté vybavenie. Poďakovanie patrí taktiež vedúcemu diplomovej práce pánovi Ing. Miroslavovi Balíkovi, Ph.D. za odborné vedenie, konzultácie, trpezlivosť a podnetné návrhy k práci.

Brno

.....

podpis autora

POĎAKOVANIE

Výzkum popísaný v tejto diplomovej práci bol realizovaný v laboratóriách podporených projektom SIX; registračné číslo CZ.1.05/2.1.00/03.0072, operačný program Výzkum a vývoj pro inovace.

Brno

.....
podpis autora

OBSAH

Úvod	13
1 Systém inteligentnej dopravy	15
1.1 Komunikačný model pre Severnú Ameriku podľa IEEE1609.x	17
1.1.1 Fyzická a linková vrstva	18
1.1.2 Sieťová a transportná vrstva	18
1.1.3 Relačná, prezentačná a aplikačná vrstva	19
1.1.4 Testovacie prevádzky a aktuálne nasadenie	19
1.2 Komunikačný model pre Európu podľa ETSI ITS-G5	21
1.2.1 Fyzická a linková vrstva	21
1.2.2 Sieťová a transportná vrstva	22
1.2.3 Relačná, prezentačná a aplikačná vrstva	22
1.2.4 Testovacie prevádzky a aktuálne nasadenie	23
1.3 Porovnanie sieťových modelov	24
1.4 Budúcnosť systému inteligentnej dopravy	25
2 Vyhradené komunikácie s krátkym dosahom	27
2.1 Správa popisujúca mapu križovatky	28
2.1.1 Element križovatky pruh	29
2.1.2 Element pruhu bod	29
2.1.3 Element atribútov pruhu	30
2.1.4 Element prepojenia pruhov	30
2.2 Správa popisujúca fázy a stavy svetelných signálov	30
2.2.1 Stav pohybu	31
2.2.2 Udalosti pohybu	32
2.3 Správa pre oznamovanie hazardných situácií	32
3 Návrh a implementácia aplikácie pre definovanie križovatiek	34
3.1 Požiadavky na aplikáciu	34
3.2 Výber implementačných prostriedkov	35
3.3 Serverová časť programu	36
3.3.1 Modul infraštruktúrneho bodu	37
3.3.2 Generovanie správ geometrie križovatky	39
3.3.3 Generovanie správ pre signály a fázy križovatky	40
3.3.4 Sprostredkovanie klientskej časti programu	43
3.3.5 Adaptér pre V2X knižnicu	44
3.3.6 Synchronizácia svetelných signálov zo semaforov	44

3.4	Klientská časť programu	46
3.4.1	Aplikačné rozhranie Google máp	47
3.4.2	Hlavný modul aplikácie	47
3.4.3	Modul pre vytvorenie geometrie križovatky	47
3.4.4	Modul pre vytvorenie signálov a fáz križovatky	48
3.4.5	Modul správ hazardných situácií	49
3.4.6	Moduly pomocných funkcií a ďalšie	49
3.4.7	Serializácia a ukladanie definovaných parametrov	50
3.5	Užívateľské rozhranie programu	50
3.5.1	Definovanie správ popisujúce mapu križovatky	51
3.5.2	Definovanie svetelných signálov a fáz križovatky	52
3.5.3	Definovanie hazardných situácií	53
4	Technická špecifikácia emulačnej jednotky	54
5	Testovanie vyvinutej aplikácie	56
5.1	Automatické testovanie zdrojového kódu	56
5.2	Verifikácia vysielaných správ	58
5.3	Testovanie funkcionality aplikácie	58
5.3.1	Testovanie v laboratóriu	58
5.3.2	Testovanie v pilotnej prevádzke	59
6	Záver	61
	Literatúra	63
	Zoznam symbolov, veličín a skratiek	65
	Zoznam príloh	67
A	Infraštruktúrna jednotka	68
B	Užívateľské rozhranie	70
C	Príklady konfigurácií	73
D	Obsah priloženého CD	76

ZOZNAM OBRÁZKOV

1.1	Rôzne varianty komunikácií v systémoch C-ITS	16
1.2	Komunikačný model IEEE1609.x WAVE.	17
1.3	Komunikačný model ETSI ITS-G5	21
2.1	Štruktúra správy MAP.	28
2.2	Štruktúra správy SPAT.	31
2.3	Štruktúra správy RSA.	32
3.1	UML diagram serverovej časti programu.	37
3.2	Vývojový diagram pre metódu <i>update</i>	42
3.3	UML diagram klientskej časti programu.	46
3.4	Grafické rozhranie pre definovanie MAP správ.	51
3.5	Grafické rozhranie pre definovanie SPAT správ.	52
3.6	Grafické rozhranie pre definovanie RSA správ.	53
4.1	Stavebné bloky RSU jednotky.	54
5.1	RSU a OBU jednotky použité pre testovanie.	59
5.2	Zoradené testovacie vozidlá (prevzaté z[15]).	60
A.1	Infraštruktúrna jednotka RSU.	68
A.2	Procesorová časť RSU jednotky.	69
A.3	RF časť RSU jednotky.	69
B.1	Grafické rozhranie pre nastavenie scenára.	70
B.2	Grafické rozhranie pre definovanie MAP správach.	70
B.3	Grafické rozhranie pre priradenie udalosti k prepojeniam.	71
B.4	Grafické rozhranie pre špecifikovanie udalostí v SPAT správach.	71
B.5	Grafické rozhranie pre priamu modifikáciu udalostí v SPAT správach.	72
B.6	Upozornenie na nesprávne nastavenie.	72

ZOZNAM TABULIEK

1.1	Porovnanie komunikačných modelov	24
5.1	Výsledky testov výstražných správ.	59

ZOZNAM VÝPISOV

5.1	Výsledky jednotkových testov.	57
C.1	Príklad konfigurácie pre priame nastavenie stavu pohybu.	73
C.2	Zakladná štruktúra konfiguračného súboru.	73
C.3	Štruktúra konfigurácie pre RSA modul.	73
C.4	Skrátená štruktúra konfigurácie pre SPAT modul.	74

ÚVOD

V dnešnej dobe (2018) sú vozidlá oveľa dostupnejšie a zvýšená prevádzka na pozemných cestných komunikáciách vedie k zvýšeniu nehodovosti, čo môže byť spôsobené viacerými skutočnosťami. Príkladom môže byť porušovanie dopravných predpisov, neprispôsobenie vedenia vozidla poveternostným podmienkam, zlý technický stav vozidla a ďalšie faktory, ktoré ovplyvňujú bezpečnosť a plynulosť cestnej premávky.

Práve pre zníženie nehodovosti a minimalizovanie strát na ľudských životoch spôsobených v dopravných nehodách sa začali vyvíjať systémy, ktoré umožnia vozidlám medzi sebou komunikovať. Vďaka tejto komunikácii si vozidlá vymenia potrebné informácie, na základe ktorých bude možné upozorniť vodičov na prípadnú kolíziu. Tým sa vodičom sprístupní dostatočná časová rezerva, ktorá umožní včasnú reakciu vodiča a možnosť zabrániť prípadnej kolízii. Vďaka týmto inováciám sa minimalizujú straty na ľudských životoch spôsobených v dopravných nehodách.

Technológie komunikácií vozidiel nadobudnú svoj plný potenciál, až v prípade, keď každé vozidlo bude vybavené prostriedkami pre komunikáciu s ostatnými vozidlami. Keďže výmena starších vozidiel, ktoré týmito komunikačnými prostriedkami nedisponujú, potrvá možno aj desaťročie. Preto sa výrobcovia sústreďujú na vytvorenie infraštruktúry, ktorá prinesie výhodu novým vozidlám. Vodiči potom môžu od infraštruktúry dostávať rôzne informácie, príkladom môžu byť navigačné informácie pre rýchly, efektívny a plynulý prejazd častí miest, kde nie sú vytvorené dopravné zápchy. Ďalší príklad môže byť prejazd cez križovatku, ktorá je riadená svetelnou signalizáciou, kedy sa informácie o aktuálnych a budúcich svetelných stavoch vysielajú infraštruktúrou. Na základe týchto informácií si vozidlo určí, či môže bezpečne takouto križovatkou prejsť, ak nie, vodič vozidla bude na túto skutočnosť upozornený. Vozidlo môže taktiež určiť optimálnu rýchlosť, ktorou bude možné daný úsek prejsť bez zastavenia.

Na základe uvedených skutočností vznikla požiadavka pre vytvorenie testovacích prostriedkov, ktoré umožnia emulovať funkcionality infraštruktúrneho bodu. Pod pojmom infraštruktúrny bod si je možné predstaviť každé zariadenie, ktoré umožňuje vysielать správy pre systém inteligentnej dopravy a je súčasťou stacionárnej inštalácie popri cestnej komunikácii. Najjednoduchší prípad môže byť dopravné značenie, ktoré vysielá výstražné správy, napríklad obmedzenie rýchlosti.

V tejto diplomovej práci je popísaný systém inteligentnej dopravy ako celok a aké sú očakávania od tohoto systému. Ďalej práca popisuje komunikačné technológie, na ktorých systém inteligentnej dopravy stavia, ich rozdiely, spôsob komunikácie, základné služby, ktoré komunikačné technológie ponúkajú. Následne sú popísané vybrané štandardizované správy a ich dátových štruktúrach, ktoré sa využívajú pre výmenu špecifických informácií medzi účastníkmi komunikácie. Vymieňané správy

sú rôznych typov, ktoré primárne zaručujú bezpečnosť a plynulosť cestnej premávky.

Hlavným cieľom tejto práce je vytvoriť testovaciu aplikáciu, ktorá bude môcť emulovať správanie infraštruktúrneho bodu so zameraním na definíciu geometrie križovatky, fázy svetelných signálov semaforov a definíciu hazardných situácií pre systém inteligentnej dopravy. Aplikácia túto funkcionality užívateľovi umožní pomocou grafického rozhrania. Grafické rozhranie sprostredkúva rýchly a efektívny spôsob vytvorenia komplexných správ pre systém inteligentnej dopravy. Užívateľom vytvorené správy sú následne kódované a pomocou infraštruktúrneho bodu poslané do bezdrôtového prostredia inteligentného transportného systému.

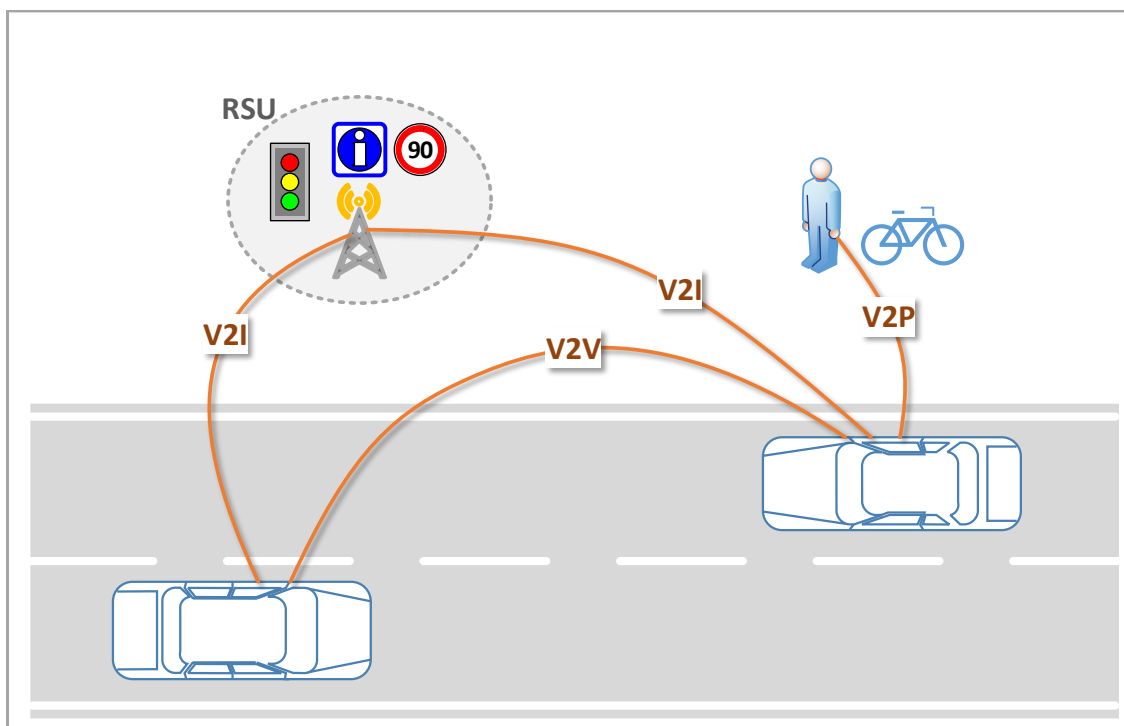
1 SYSTÉM INTELIGENTNEJ DOPRAVY

Pod názvom Systém inteligentnej dopravy ITS (Intelligent Transport Systems) sa skrýva viacero rôznych technológií. V dnešnej dobe (2018) známe pasívne systémy pre sledovanie vyťaženia pozemných komunikácií. Dáta zozbierané z týchto systémov slúžia ako vodítko pri rozhodovaní budovania nových cestných komunikácií pre zrýchlenie prejazdu daných úsekov a ďalších optimalizácií. Podobnými pasívnymi systémami sú rôzne distribuované systémy používané v mestách, kde na základe dlhodobo zozbieraných informácií o vyťažení cestných komunikácií, sa upravujú dĺžky stavov svetelnej signalizácie naprieč celým mestom a tým zabezpečujú plynulý prejazd mestom. Jedným príkladom riešenia takéhoto systému, ktorý stavia na princípoch ITS, je popísaný v článku [17]. Tento systém je navrhnutý pre spracovanie a rýchlu konvergenciu z dát získaných v reálnom čase.

Ďalšou veľkou skupinou ITS sú kooperatívne systémy, ktoré sa označujú skratkou C-ITS (Cooperative ITS). Pre komunikáciu systémov C-ITS bolo vyhradené rádiové pásmo 5,9 GHz. O systémoch C-ITS možno hovoriť ako o aktívnych, pretože pri vytváraní informácií o aktuálnej situácii na cestnej komunikácii vytvárajú priamo účastníci cestnej premávky. Účastníkom cestnej premávky je v kontexte systémov C-ITS myslené napríklad motorové vozidlo s prostriedkami, ktorými môže pristupovať k systému C-ITS. Prostriedok, ktorý sprístupní komunikáciu vozidla zo systémom C-ITS je palubná jednotka OBU (On Board Unit). OBU má za úlohu vysielat údaje, ktoré sa používajú v situáciách pre zaistenie bezpečnosti posádky vozidla. V prípade prijímania tieto informácie spracovať a v adekvátnej forme zobrazit vodičovi relevantné upozornenia. Popis variant komunikácie v systémoch C-ITS, ktoré sú zobrazené na obrázku 1.1:

- vozidlo k vozidlu V2V (Vehicle to Vehicle), prípadne C2C (Car to Car), jedná sa o komunikáciu medzi vozidlami, na základe ktorej fungujú antikolýzne protokoly, ktoré využívajú práve OBU,
- vozidlo ku chodcovi V2P (Vehicle to Pedestrian), táto komunikácia opäť združuje niekoľko rôznych situácií, kde môže dôjsť k ohrozeniu chodcov na prechodoch a popri cestách, najmä za zníženej viditeľnosti,
- vozidla k infraštruktúre V2I (Vehicle to Infrastructure) je použitá najmä pre získavanie rôznych informácií o prípadných hazardoch a hrozbách, ktoré môžu skomplikovať prejazd úsekom cestnej komunikácie.

Príkladom môžu byť informácie o stave vozovky (ľad na ceste), prekážkach na ceste, o aktuálnych a budúcich stavoch svetelnej signalizácie križovatiek. Infraštruktúru tvoria k tomu určené jednotky, ktoré sú strategicky rozmiestnené popri cestách. Takáto jednotka sa označuje ako infraštruktúrny bod RSU (Road Side Unit). RSU jednotka je typ vstavaného počítača, ktorý má hardvérové a softvérové vybavenie



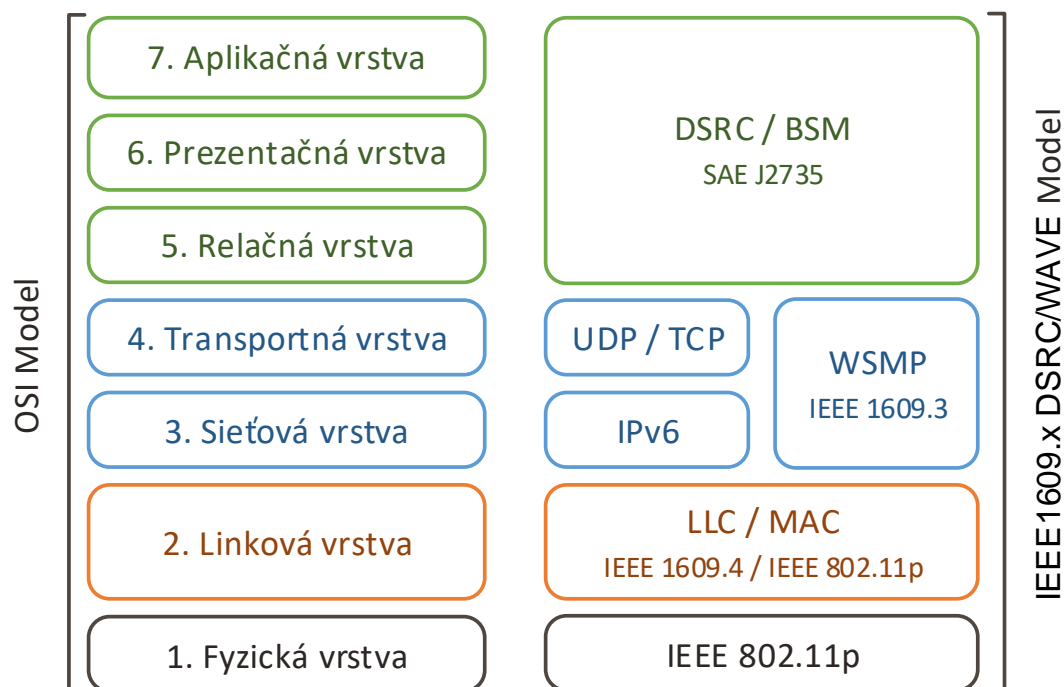
Obr. 1.1: Rôzne varianty komunikácií v systémoch C-ITS

pre komunikáciu v systémoch C-ITS. Princípy komunikácií v systémoch C-ITS sú popísané v kapitolách 1.1 a 1.2. Štruktúry komplexných správ, ktoré bude RSU jednotka vysielat sú popísané v kapitole 2. Aplikácia, ktorá užívateľovi umožní vytváranie správ, je podrobne popísaná v kapitole 3.

V systémoch C-ITS existuje ešte viacero ďalších typov komunikácií, ktoré označujú účastníkov alebo účel komunikácie. Pre všetky typy komunikácií sa používa označenie vozidlo ku všetkému V2X (Vehicle to Everithing). Viac informácií možno nájsť v článku [22].

1.1 Komunikačný model pre Severnú Ameriku podľa IEEE1609.x

IEEE1609.x je rodina štandardov pre bezdrôtový prístup v automobilovom prostredí WAVE (Wireless Access in Vehicular Environments). Model je prispôsobený k požiadavkám systému V2X, ktoré sú z pohľadu fyzickej a linkovej vrstvy modelu IOS/OSI rýchly prístup k prenosovému médiu s dostatočnou prenosovou rýchlosťou (maximálne 27 Mbit/s, ale používajú sa rýchlosti 6 Mbit/s a 10 Mbit/s). Na sieťovej a transportnej vrstve WAVE implementuje protokol IPv6 a k nemu transportné protokoly UDP/TCP¹. Ďalším novým protokolom, zastupujúcim sieťovú a transportnú vrstvu, je protokol krátkych správ WSMP (WAVE Short Message Protocol), ktorý je popísaný v kapitole 1.1.2.



Obr. 1.2: Komunikačný model IEEE1609.x WAVE.

Vyššie vrstvy definujú služby a aplikácie, ktoré zaistujú požiadavky systému V2X [14], [1]. Na obrázku 1.2 je zobrazený WAVE komunikačný model a prirovnaný k referenčnému modelu ISO/OSI.

¹Pri prenose informácií pomocou IPv6 sa predpokladá použitie transportného protokolu UDP pretože krátka dĺžka spojenia neumožní nadviazanie obojsmernej komunikácie pomocou TCP.

1.1.1 Fyzická a linková vrstva

Najnižšie vrstvy modelu sú postavené na štandarde IEEE 802.11p, ktorý je odvodený z pôvodného štandardu IEEE 802.11a. Podstatným rozdielom štandardu IEEE 802.11p od ostatných je, že povoľuje prácu mimo kontext BSS (Basic Service Set), čiže stanice sa pred prenosom nemusia autentifikovať a autorizovať. Vďaka tomu je čas výmeny informácií veľmi rýchly [12](kapitola 5.2.10). Komunikácia prebieha v pásme 5,85–5,92 GHz, ktoré je rezervované pre ITS aplikácie. Toto pásmo je rozdelené na sedem kanálov so šírkou pásma 10 MHz. Z týchto kanálov sa vždy použijú minimálne dva, pričom jeden sa nazýva kontrolný a používa sa primárne pre prenos správ o bezpečnosti BSM (Basic Safety Message). Správa BSM obsahuje informácie o polohe, rýchlosti a smere pohybu vozidla. Druhý sa nazýva servisný a je určený k prenosu dát rôznych služieb, ktoré sú definované na aplikačnej úrovni [14](kapitola 5.6). Z pohľadu prenosu môže zariadenie disponovať jedným alebo dvoma fyzickými rozhraniami. Zariadenie s jedným fyzickým rozhraním prepína medzi servisným a kontrolným kanálom v definovanom intervale. S pribúdajúcou hustotou komunikácie na médiu zariadenie prestane prepínať medzi kanálmi a zostane komunikovať len na kontrolnom kanáli, pre príjem BSM správ. Zariadenie s dvoma fyzickými rozhraniami vždy komunikuje na oboch kanáloch súčasne [13](kapitola 5).

1.1.2 Sieťová a transportná vrstva

Na sieťovej a transportnej vrstve komunikačný model implementuje IPv6 spolu s UDP/TCP protokolmi, tak ako sú známe z klasických IP sietí. Novinkou je ale protokol krátkych správ WSMP, ktorý zastrešuje jak sieťovú tak transportnú vrstvu. Princíp komunikácie je na základe požiadaviek a odpovedí. Požiadavky sú rozdelené do typov napríklad *poskytovateľ*, *užívateľ*, *WAVE krátka správa* atď. *Poskytovateľ* a *užívateľ* majú možnosť služby vyšších vrstiev ohlasovať a tak distribuovať informácie ďalej cez sieť. WSMP taktiež zaisťuje, že koncovému užívateľovi sa zobrazia informácie, ktoré sú relevantné a ku ktorým sa koncový užívateľ prostredníctvom aplikácie prihlásil. Tieto informácie sa uchováajú v tabuľkách (napríklad *Provider-ServiceRequestTable*) ktoré sú súčasťou protokolu WSMP.

Stále je však potrebné rozlíšiť dáta aplikácií a služieb, toto je umožnené pomocou identifikátorov PSID (Provider Service Identifier), ktoré sú registrované k jednotlivým službám a aplikáciám. Pridelené hodnoty PSID identifikátorov sú uvedené v štandarde IEEE 1609.12 [14](kapitola 5.9).

1.1.3 Relačná, prezentačná a aplikačná vrstva

Z pohľadu vyšších vrstiev sú na výber dva komunikačné scenáre, jeden *inzerovaný* a *neinzerovaný*. Inzerovaný spôsob komunikácie existuje medzi tzv. *poskytovateľom* a *užívateľom*. Poskytovateľ posiela správy (inzercie) o službách, ktoré poskytuje, k službám sú priradené ich PSID identifikátory a číslo servisného kanálu, na ktorom sú služby dostupné. Po prijatí tejto správy užívateľom si užívateľ zistí či sa nachádza v konvergentnej oblasti správy, ak áno môže sa pripojiť k ponúkaným službám a v rámci konvergentnej oblasti preposlať informácie o službách ďalším užívateľom, pokiaľ nevyprší platnosť dát, ktoré sú obsiahnuté v správe [14](kapitola 5.7). Konvergentná oblasť môže byť väčšia ako je dosah použitej technológie na fyzickej vrstve, preto sa sprostredkovaná komunikácia zaviedla. Na tomto princípe funguje manažér prostriedkov RM (Resource Manager), ktorý sprostredkováva aplikáciám prístup k nižším vrstvám a vytvára potrebné správy pre nižšie vrstvy. V prípade OBU RM filtruje iba žiadúci obsah pre koncovú aplikáciu [11](kapitola 1).

Neinzerovaný spôsobom komunikácie je typický ad-hoc prenos dát, koncové uzly nemajú predurčenú úlohu (poskytovateľ, užívateľ, ...). Tento spôsob sa používa pre prenos správ o bezpečnosti BSM. Správy BSM sú vysielané v periodických intervaloch, alebo vyvolané udalosťou (napríklad prudké brzdenie). Sada správ využívajúci tento spôsob komunikácie je súčasťou štandardu SAE J2735. Niektoré z nich sú popísané v kapitole 2.

1.1.4 Testovacie prevádzky a aktuálne nasadenie

Ministerstvo dopravy Spojených Štátov Amerických dňa 1. 8. 2016 spustilo dvadsať mesačnú fázu vývoja a nasadenia technológií V2X. V tejto fáze boli spustené pilotné projekty, ktoré zabezpečujú nasadenie a testovanie aplikácií pre systémy C-ITS.

Pilotný projekt WYDOT (Wyoming Department of Transportation) má za úlohu otestovať aplikácie, ktoré zvýšia bezpečnosť premávky za nepriaznivého počasia, primárne pre nákladnú prepravu:

- varovanie pred čelnou zrážkou v neprehľadných situáciách, systém však neprevezme riadenie vozidla, aby zabránil zrážke,
- varovanie a informácie sprostredkované infraštruktúrou, príkladom sú práca na ceste, zladovatená vozovka, odporúčaná rýchlosť, voľné parkovacie miesta a ďalšie.

Pre tento projekt bola vytvorená infraštruktúra pozostávajúca zo sedemdesiatimi piatimi RSU jednotkami a štyristo testovacích vozidiel bolo vybavených palubnou jednotkou (OBU). Štart projektu je plánovaný na zimu roku 2017. Koncom roku 2018 sa projekt rozšíri aj o testovanie s komerčnými vozidlami a bude trvať do zimy roku 2019.

Projekt NYCCVP (New York City Connected Vehicle Project) je zameraný na časti mesta New York s vysokou nehodovosťou, kde pomocou technológií V2X zvyšujú bezpečnosť a plynulosť premávky. Projekt sa zameriava na testovanie rôznych varovaní. Príklad niektorých varovaní a správ:

- asistent prejazdu križovatkou,
- varovanie o aute v mŕtvom uhle,
- varovanie o zmene pruhu,
- chodec na prechode.

Vytvorená infraštruktúra pozostáva z viac ako štyristo RSU jednotkami, z ktorých je väčšina nainštalovaná ako súčasť svetelnej signalizácie križovatiek. Palubnými jednotkami bolo vybavených viac ako osem tisíc vozidiel rôznych typov (taxíky, autobusy, vozidlá zásielkových služieb, atď.). Tento projekt je rozdelený do troch fáz. Prvá fáza bola plánovacia a začala v Septembri 2015. Druhá fáza začala v septembri 2016, kedy sa začal vývoj a integrácia potrebných prostriedkov pre C-ITS a zber dát z prvých prototypov. Spustenie projektu je plánovaný na Júl 2018. Viac informácií o projekte možno nájsť na [18].

Posledným pilotným projektom je THEA (Tampa Hillsborough Expressway Authority), ktorý je zameraný na zvyšovanie bezpečnosti a plynulosti premávky na rýchlostných cestách a diaľniciach. Tento projekt sa tiež zameriava na už spomínané varovania. Ďalšie varovania:

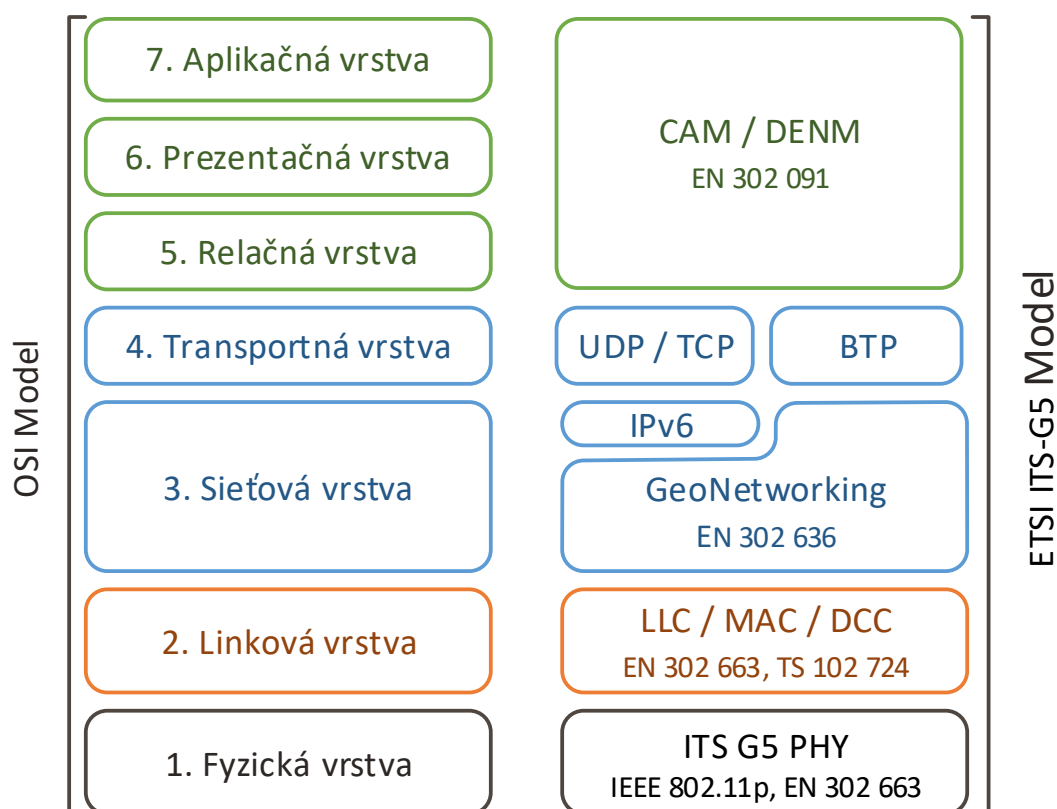
- odporúčaná dojazdová vzdialenosť ku kolóne vozidiel,
- vjazd do zakázaného pruhu,
- priorita prejazdu mestskej hromadnej dopravy,
- chodec na prechode.

Tento projekt primárne využíva viac ako dvetisíc osobných vozidiel pre testovanie rôznych situácií a pripája viac ako päťsto chodcov pomocou mobilných aplikácií, ktorými možno simulovať oveľa väčší počet situácií, ktoré môžu nastať v systémoch C-ITS.

Aktuálne informácie o projektoch a nasadení technológie V2X je možno nájsť na webových stránkach ministerstva dopravy Spojených Štátov Amerických [23].

1.2 Komunikačný model pre Európu podľa ETSI ITS-G5

Európsky ústav pre telekomunikačné normy ETSI (European Telecommunications Standards Institute) je organizácia, ktorá vyvinula komunikačný model pre Európu. Štandardy ETSI sú voľne dostupné na webovej stránke inštitútu. Komunikačný model je popísaný v štandarde ETSI EN 302 665 [6]. Nižšie vrstvy komunikačného modelu sú opäť postavené na štandarde IEEE 802.11p, sieťová a vyššie vrstvy sú oproti predošlému modelu rozdielne (obrázok 1.3) [22].



Obr. 1.3: Komunikačný model ETSI ITS-G5

1.2.1 Fyzická a linková vrstva

Fyzická a linková vrstva stavia na princípoch komunikačného modelu určeného pre Severnú Ameriku, ktorý je popísaný v kapitole 1.1.1. Štandard ETSI bližšie špecifikuje vysielacie výkony a prenosové rýchlosti v jednotlivých kanáloch technológie IEEE 802.11p. Kanály sú rozdelené do skupín ITS-G5A, ITS-G5B a ITS-G5D [7].

- ITS-G5A je určená pre aplikácie, ktoré zaistujú bezpečnosť, používa tri kanály, pričom jeden z nich je kontrolný,
- ITS-G5B zastrešuje ostatné aplikácie, ale zatiaľ sa nevyužíva,
- ITS-G5D je vyhradená pre budúce použitie,
- ITS-G5C táto skupina má vyhradené pásmo (5,47 GHz–5,725 GHz) a teda nespadá do pásma technológie IEEE 802.11p, v tomto pásme sa nepracuje mimo kontext BSS a jeho použitie je cielené pre prenos multimediálnych dát [10].

Pre prácu a správu kanálov v skupinách ITS-G5A, ITS-G5B bola vytvorená pod-vrstva decentralizovanej kontroly preťaženia DCC (Decentralized Congestion Control). DCC monitoruje vyťaženie kanálov a realizuje stavový automat pozostávajúci z troch stavov *pokojný*, *aktívny* a *obmedzený*. Každý stav odpovedá miere vyťaženia kanálu. Na základe stavov automatu sa upravujú priority pre odosielané správy. V prípade najživšieho vyťaženia sa posielajú len správy týkajúce sa bezpečnosti. Podrobný popis činnosti DCC možno nájsť v štandarde [9].

1.2.2 Sieťová a transportná vrstva

V prípade európskeho komunikačného modelu sa opäť na sieťovej a transportnej vrstve implementuje protokol IPv6 s UDP/TCP transportnými protokolmi. Novinkou sú protokoly GeoNetworking a základný transportný protokol BTP (Basic Transport Protokol). GeoNetworking je smerovací protokol, ktorý je optimalizovaný pre prenos a adresovanie v prostredí ad-hoc sietí. Umožňuje rozposielať pakety bez predošlého nadviazania a monitorovania logického komunikačného spojenia a ďalšie pokročilejšie metódy smerovania paketov. Adresovanie a smerovanie je sprostredkované na základe geografických pozícií alebo oblastí. Podobne ako pri IPv6, GeoNetworking protokol podporuje viacero typov prenosu a to geo-broadcast, geo-unicast, geo-anycast a broadcast na jeden skok. Použitie typu prenosu potom závisí na aplikácii a charaktere dát, ktoré sa snaží preniesť [8], [10].

Protokol BTP sa používa pre multiplexovanie aplikačných dát. BTP definuje dva typy hlavičiek, jedna z nich používa zdrojový a cieľový port (ekvivalent UDP). Druhá hlavička sa používa pri jednostrannej komunikácii a obsahuje len cieľový port a pole ďalších informácií [5].

1.2.3 Relačná, prezentačná a aplikačná vrstva

Funkcie relačnej, prezentačnej a z časti aplikačnej vrstvy zastupuje vrstva služieb, alebo vybavenia. Vrstvu služieb zastupujú CAM (Cooperative Awareness Messages) a DENM (Decentralized Environmental Notification Message). CAM združuje informácie o veľkosti a pozícii vozidla, smere a rýchlosti pohybu, ktoré následne periodicky vysiela do svojho blízkeho okolia. Na základe informácií v správe CAM,

účastníci cestnej premávky môžu predchádzať kolíziám. Správa CAM je ekvivalentom správy BSM z Amerického komunikačného modelu. Služba CAM je popísaná v štandarde ETSI EN 302 637-2. DENM je služba, ktorá má za úlohu informovať účastníkov cestnej premávky o prípadných hazardoch, ktoré sa môžu vyskytnúť v smere ich cesty. Správy DENM sú vysielané na základe udalostí a pomocou GeoNetworking protokolu môžu byť správy vysielané do väčšej geografickej oblasti ako je reálny dosah technológie IEEE 802.11p. Služba DENM je popísaná v štandarde ETSI EN 302 637-3 [22].

1.2.4 Testovacie prevádzky a aktuálne nasadenie

Pre účely nasadenia a testovania technológií V2X postavené na komunikačnom modeli ETSI ITS-G5 v Európe vznikla platforma C-ROADS. Platforma C-ROADS má za úlohu koordinovať ďalšie pilotné projekty zamerané na špecifické aspekty systémov C-ITS zo zameraním na infraštruktúru. K tejto platforme sa v novembri roku 2017 pripojila aj Česká republika na základe čoho sa začnú vytvárať testovacie infraštruktúry vo vybraných mestách a diaľniciach [2].

Ďalším projektom na testovanie a nasadenie technológií V2X je InterCor [15], ktorý spadá pod platformu C-ROADS. Projekt InterCor je zameraný na vývoj a optimalizáciu nasledujúcich typov aplikácií:

- signalizácia rôznych správ vo vozidlách,
- varovanie o prácach na ceste,
- asistent optimálnej rýchlosti v rôznych situáciách,
- signalizácia rôznych správ vo vozidlách,
- optimalizácia viacnásobnej nákladnej prepravy.

InterCor koncom roku 2017 už úspešne demonštroval nasadenie a použitie V2X technológií na testovacích dňoch, kde zúčastnení mohli otestovať implementáciu a funkcionality svojich OBU jednotiek. Projekt začal 1. 9. 2016 a ukončenie projektu je plánované na 31. 8. 2019.

Projekt ETPC (European Truck Platooning Challenge) predstavuje koncept využitia systémov C-ITS pre zefektívnenie vedenia nákladných vozidiel, kde je prvé vozidlo označené ako vedúce a ostatné vozidlá sú zretazené s časovým odstupom za vedúcim. Zretazené vozidlá upravujú rýchlosť podľa vedúceho a to tak aby minimalizovali nutnosť brzdenia a následného zrýchľovania. Vďaka tomuto prístupu sa šetria pohonné hmoty, minimalizuje sa znečistenie ovzdušia a zlepši sa prejazdnosť miest. Štart tohoto projektu je plánovaný na rok 2018 [4].

1.3 Porovnanie sieťových modelov

Nasledujúca tabuľka ukazuje prehľad použitých technológií v oboch popisovaných modeloch v zhlľadom k modelu ISO/OSI.

Tab. 1.1: Porovnanie komunikačných modelov

ISO/OSI	IEEE 1609.x WAVE	ETSI ITS-G5
Aplikačná	SAE DSRC, BSM	CAM, DENM
Prezentačná		
Relačná		
Transportná	WSMP	BTP
Sieťová		GeoNet
Linková	LLC, IEEE 802.2	
	IEEE 1609.4	DCC
	Multi-Channel Operation	Decentralized Congestion Control
Fyzická	IEEE 802.11p	

Oba komunikačné modely na fyzickej vrstve vychádzajú zo štandardu IEEE 802.11p, ktorý povoľuje prácu mimo kontext BSS. Americký model pre komunikáciu využíva všetkých sedem kanálov, pričom štvrtý kanál je použitý ako kontrolný a ostatné ako servisné kanály. Európsky model rozdeľuje týchto sedem kanálov do troch skupín. Zatiaľ sa využíva len prvá skupina, v ktorej sa používajú tri kanály a jeden z nich je kontrolný. Oba komunikačné modely obmedzujú maximálne vysielačie výkony v jednotlivých kanáloch. Vysielačie výkony medzi Európskym a Americkým komunikačným modelom nie sú rovnaké. Je to dané predovšetkým iným využitím jednotlivých kanálov a tiež preto, že Európsky model zatiaľ zásadne limituje vysielačie výkony v nevyužitých kanáloch.

Pre prácu komunikačných modelov mimo kontext BSS sú na linkovej vrstve vytvorené podporné mechanizmy, ktoré zaisťujú prenos dát vo viacerých kanáloch, ich prioritu a kvalitu služieb. Pre americký komunikačný model sú tieto mechanizmy popísané v štandarde IEEE 1609.4, ktorý ma za úlohu jednotlivé dátové prenosy podľa rôznych požiadaviek na akútnosť prenášaných dát. Pre zaistenie kvality služieb a prístupu k médiu sa používajú princípy popísané v štandarde IEEE 802.11e (EDCA). Európsky komunikačný model pre prenos dát vo viacerých kanáloch používa decentralizovanú kontrolu preťaženia (DCC). DCC implementuje stavový automat, ktorý je riadený hustotou komunikácie. Zo zvyšujúcou hustou komunikácie sa prenos dát

obmedzuje len na kontrolný kanál, kde sú vymieňané len správy zaistujúce bezpečnosť. Pre zaistenie kvality služieb a prístup k médiu je opäť použitý EDCA.

Na sieťovej a transportnej vrstve oba komunikačné modely implementujú IPv6 spolu s UDP/TCP protokolmi. Charakter komunikácie v kooperatívnych systémoch inteligentnej dopravy viedol k vytvoreniu nových sieťových a transportných protokolov. Americký model používa protokol krátkych správ (WSMP), ktorý je navrhnutý pre prenos dát s vysokou prioritou a malou latenciou. Európsky model používa smerovací protokol GeoNetworking, ktorý dokáže distribuovať správy do geografických oblastí bez nutnosti monitorovania aktívneho spoja k tejto oblasti. Pre prenos správ sa používa jednoduchý transportný protokol BTP.

Protokoly vyšších vrstiev sú reprezentované ako typ jedno či viac účelovej správy. Z oboch modelov správy BSM a CAM plnia rovnaký účel a to vymieňanie informácií za účelom zaistenia bezpečnosti posádky vozidla. Americký model používa pre prenos ďalších rôznych informácií súbor správ DSRC, ktorého niektoré časti sú popísané v nasledujúcej kapitole. V Európskom komunikačnom modeli sa pre prenos ďalších informácií využíva decentralizované hlásenie o environmentálnom upozornení (DENM). Doručovanie DENM správ stavia na princípoch GeoNetworking protokolu a možnosti adresovania geografickej oblasti. Správy slúžiace pre posielanie informácií o geometrii križovatky a stavoch svetelnej signalizácie sú pre oba komunikačné modely rovnaké.

1.4 Budúcnosť systému inteligentnej dopravy

Za posledné desaťročie sa rýchlo vyvíjali všetky oblasti bezdrôtových prenosov, čo viedlo k zvýšeniu prenosových rýchlostí, spoľahlivosti a zväčšeniu dosahov. V oblasti mobilných technológií bol vyvinutý systém LTE (Long Term Evolution), ktorý vo svojom štrnástom vydaní podporuje komunikácie v automobilovom prostredí [20]. Táto technológia sa označuje skratkou C-V2X² (Cellular V2X), opäť je použité pásmo 5,9 GHz, a podporuje priamu komunikáciu medzi zariadeniami, pričom sú stále zachované služby mobilných operátorov. Výhoda C-V2X oproti IEEE 802.11p je zvýšenie dosahu a priepustnosti. C-V2X taktiež lepšie potlačuje dopplerov jav pri komunikácii rýchlo sa pohybujúcich komunikačných zariadení. Potencionálna úspora hardvérových prostriedkov, keďže v súčasnosti sú moderné vozidlá už bežne vybavené LTE modulami, ktoré sú využité napríklad systémom eCall (emergency call), ktorý v prípade detekovania nehody automaticky vyvolá tiesňové volanie. Podrobné porovnanie C-V2X a IEEE 802.11p DSRC možno nájsť v článku [19].

²Nezamieňať s C-ITS, čo je kooperatívny systém inteligentnej dopravy.

Napriek tomu, že technológia C-V2X sa na prvý pohľad zdá ako lepšia alternatíva ku klasickej V2X, má niekoľko nevýhod. Keďže špecifikácia pre C-V2X bola vydaná v roku 2017, použiteľné hardvérové moduly je možné očakávať v časovom horizonte 3 až 5 rokov, čo výrazne oneskorí nasadenie V2X technológií. V súčasnej dobe (2018) stále prebiehajú rokovania o tom, ktorú z technológií použiť.

2 VYHRADENÉ KOMUNIKÁCIE S KRÁTKYM DOSAHOM

Pre systémy inteligentnej dopravy bol organizáciou SAE International vyvinutý zoznam správ, ktoré sú popísané v štandarde SAE J2735 [3]. Tento zoznam správ je označovaný skratkou DSRC (Dedicated Short Range Communications) čo v preklade znamená Vyhradené komunikácie s krátkym dosahom. Správy popísané v tomto štandarde sú určené pre použitie v Severnej Amerike.

Jedným z dôvodov vzniku tohoto štandardu je nutnosť zjednotiť štruktúru vymieňaných správ naprieč veľkému počtu výrobcov dopravných prostriedkov a rôznych systémov. Výhodou štandardizovaných správ je, že výrobcovia jednotiek systémy C-ITS nemusia štandardizovať proprietárne protokoly pre prenos dát, čo by skomplikovalo vývoj C-ITS technológií. Vďaka štandardizovaným správam budú koncové aplikácie vždy schopné reprezentovať informácie obsiahnuté v správach aj keď sú od rôznych výrobcov.

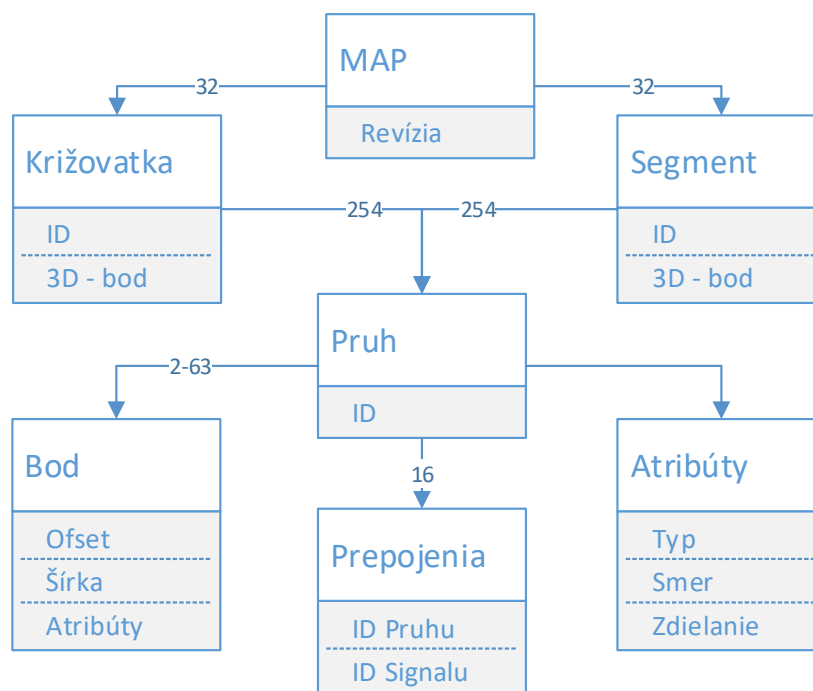
V štandarde sú správy rozdelené do troch skupín podľa priority. Skupina s najvyššou prioritou je „bezpečnosť“. Ďalšia skupina pre „mobilitu“ a posledná pre komerčné či privátne využitie.

Každá správa je navrhnutá k špecifickému požiadavku C-ITS systému. V nasledujúcom zozname sú vybrané niektoré správy a ich krátky popis:

- BSM (Basic Safety Message) správa o bezpečnosti sa používa vo viacerých prípadoch, slúži pre výmenu bezpečnostných údajov stavu vozidla,
- CSR (Common Safety Request) Spoločná požiadavka na bezpečnosť – pomocou tejto správy je možné vynútiť poslanie ďalších informácií o bezpečnosti, ktoré sú voliteľnou súčasťou správy BSM,
- ICA (Intersection Collision Avoidance) vyvarovanie kolízií na križovatkách – táto správa slúži na upozornenie ostatných vozidiel v prípade, že niektoré vozidlo prechádza cez križovatku aj napriek tomu, že nemá oprávnenie,
- MAP (Map Data) správa prenáša geografické informácie o segmente cesty, alebo celej križovatky, podrobný popis je v kapitole 2.1,
- RSA (Road Side Alert) správa je určená pre prenos informácií o hazardných situáciách, podrobný popis je v kapitole 2.3,
- SPAT (Signal Phase And Timing) správa prenáša informácie o stavoch svetelnej signalizácie križovatky, podrobný popis je v kapitole 2.2,
- TIM (Traveler Information Message) tieto správy sú určené pre posielanie informácií o dopravnom značení a ďalších textovo orientovaných správ.

2.1 Správa popisujúca mapu križovatky

Táto správa je v štandarde [3] označená skratkou MAP (Map Data). Správa MAP sa používa pre odosielanie viacerých typov geografických informácií. Jedna správa môže obsahovať definíciu až tridsiatich dvoch máp križovatiek. Definície križovatiek sú od seba v správe rozlíšené pomocou referenčného identifikátora. Dátová štruktúra križovatky je zložená z definície 3D-bodu, šírky pruhov, rýchlostných limitov, zoznam pruhov a názvu.



Obr. 2.1: Štruktúra správy MAP.

Na obrázku 2.1 je znázornená hrubá štruktúra logických prepojení v MAP správe. Elementy správy MAP sú podrobne popísané v texte nižšie. Ďalšou dôležitou súčasťou správy MAP je definícia segmentov ciest. Dátová štruktúra segmentov ciest je totožná so štruktúrou pre definíciu križovatiek. Tieto segmenty sa používajú pre vytýčenie geografických informácií, ktoré sú spojené s udalosťami typu napríklad „práca na ceste“. Tým sa umožní účastníkom cestnej premávky sprehľadniť situáciu o prejazdnosti daného úseku.

Správa MAP obsahuje množstvo voliteľných položiek. To, ktoré položky bude vo výsledku správa obsahovať, závisí na konkrétnom použití, čo znamená, že príjemca týchto správ musí byť schopný rozpoznať a interpretovať všetky typy informácií obsiahnutých v správe. Jediným povinným parametrom je poradie (revízia) správy,

ktorým možno rozlíšiť prijaté duplicitné správy od správ s rozdielnym alebo aktualizovaným obsahom.

2.1.1 Element križovatky pruh

Element pruh je kľúčový pre definíciu akejkoľvek mapy križovatky či segmentu cesty. Pruh môže interpretovať akúkoľvek súčasť cestnej premávky. Štandard definuje viacero typov pruhov napríklad pruh pre motorové vozidlá, cyklistov, prechodov pre chodcov, ostrovčekov a podobne.

V rámci definície jednej križovatky môže byť zadaných až 254 pruhov. Pruhy sú od seba odlišené číselným identifikátorom, ktorý je unikátny v rámci definície križovatky. Ďalšie dôležité súčasti pruhu sú zoznam bodov, atribúty a prepojenia, ktoré sú podrobne popísané v kapitolách nižšie. Okrem týchto súčastí je možné nastaviť povolené manévry, zoznam prekrytí pruhov a názov pruhu.

Povolené manévry určujú možnosti jedného pohybu či ich kombináciou na konci pruhu, napríklad odbočenie doprava, nutnosť zastaviť na konci pruhu a podobne. Zoznam prekrytí pruhu určuje, ktoré pruhy prekrývajú či pretínajú daný pruh v rámci jednej križovatky. Typickým príkladom pretínania je prechod pre chodcov, prekrývanie most ponad diaľnicou.

2.1.2 Element pruhu bod

Element bod je základným stavebným prvkom pruhu. Každý pruh je definovaný minimálne dvoma bodmi. V prípade, keď má daný pruh zložitejšiu geometriu, napríklad zaoblenie, je možné toto zakrivenie aproximovať sekvenciou bodov, maximálne šesťdesiatimi troma bodmi.

Každý bod nesie informáciu o polohe a to v podobe kartografických súradníc. Nevýhodou použitia kartografických súradníc je pre vyjadrenie dostatočnej presnosti, ktorá by spĺňala požiadavky ITS, nutné použiť až 32bitovú presnosť čísiel. S použitím veľkých čísiel narastá veľkosť správ čo nieje užitočná vlastnosť pri bezdrôtovom prenose. Z uvedených dôvodov sa prechádza na definovanie polôh bodov v karteziánskej súradnicovej sústave, kedy sa ako referenčný bod považuje už spomínaný 3D-bod v hlavnej štruktúre elementu križovatky. Následne je prvý bod odvodený ofsetom od 3D-bodu križovatky a ďalšie body pruhu sú odvodené ofsetom od predošlého bodu pruhu. Výhodou je, že ofsety už nie sú veľké čísla¹ a teda v dostatočnej presnosti je možné ofsety vyjadriť s menším počtom bitov a výrazne zmenšiť veľkosť celej správy.

¹Jednotky až stovky metrov, základná jednotka je centimeter.

Pre každý segment vytvorený medzi dvoma bodmi je možno nastaviť množstvo rôznych parametrov. Napríklad presnejšie špecifikovať šírku segmentu, prípadne definovať rôzne prvky špecifické k úseku, napr. obrubník na pravej strane segmentu, plná čiara na ľavej strane segmentu. Podrobný výpis všetkých atribútov možno nájsť v štandarde [3](kapitola 6.69).

2.1.3 Element atribútov pruhu

Ďalší povinný element pruhu je atribút pruhu, ktorý združuje niekoľko dôležitých údajov. Pomocou atribútov sa definuje typ, zdieľanie a smerové použitie pruhu. Smerové použitie môže byť prichádzajúca alebo odchádzajúca premávka križovatky, prípadne obojsmerné použitie. Typ môže byť napríklad vozovka, koľajová trať, chodník, parkovacia zóna a ďalšie. Po nastavení typu je nutné nastaviť zdieľanie pruhu kde je možné špecifikovať napríklad pruh pre vozidlo môže použiť cyklista, autobus či chodec.

2.1.4 Element prepojenia pruhov

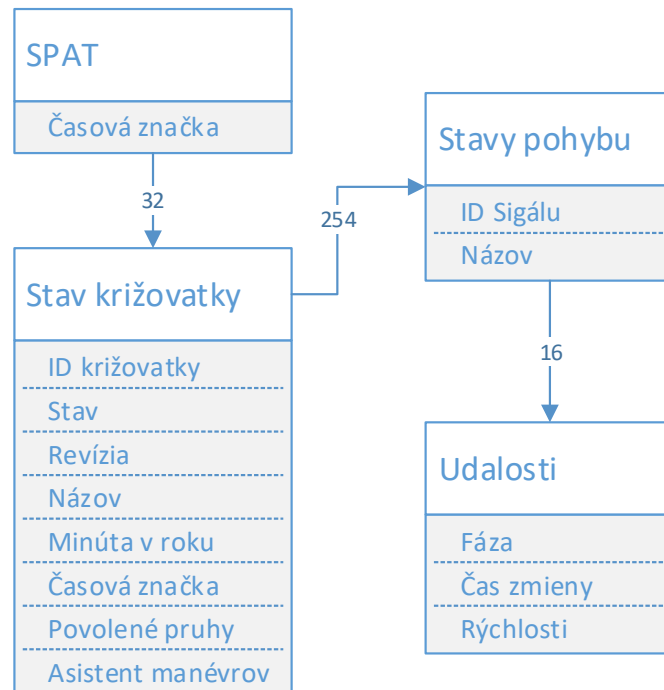
Každý pruh môže obsahovať zoznam pruhov, ku ktorým je logicky pripojený. Napríklad, z príchodzieho pruhu je možné prejsť križovatkou priamo, alebo odbočiť doprava. Prichádzajúci pruh bude teda obsahovať dve prepojenia.

Pre definíciu jedného prepojenia slúži element prepojení pruhov. Z jedného pruhu je možné vytvoriť až šesťnásť prepojení. Každé z prepojení obsahuje identifikátor pruhu, ku ktorému sa pripája. Prepojenie má tiež svoj unikátny identifikátor, pomocou ktorého sa zo správy SPAT môžu nadviazať dáta o manévroch, potrebných pre prejazd úseku prepojenia. Ďalší identifikátor, ktorý element prepojenia obsahuje je identifikátor skupiny signálov, ktorý priraduje fázy svetelných signálov definované v SPAT správe k jednotlivým prepojeniam.

2.2 Správa popisujúca fázy a stavy svetelných signálov

Správa o fázach a časovaní signálov je v štandarde [3] označovaná skratkou SPAT (Signal Phase And Timing). Táto správa spolu zo správou MAP tvorí kompletnú informáciu o danej križovatke a jej fázach a stavoch svetelnej signalizácie. Podobne ako správa MAP, správa SPAT môže preniesť fázy a stavy signálov až pre tridsaťdva križovatiek naraz. Na obrázku 2.2 je znázornená hrubá štruktúra správy SPAT. Definícia stavu jednej križovatky sa skladá z viacerých povinných parametrov. Opäť unikátny identifikátor, ktorý sa tentoraz skladá z identifikátoru križovatky, pre

ktorú je definícia stavu určená a identifikátoru správcu cestných komunikácií. Poradie (revízia) má rovnakú úlohu ako pri definícii križovatky v správe MAP. Ďalším parametrom je stav, pomocou ktorého je možno definovať stav riadiacej jednotky pre svetelnú signalizáciu.



Obr. 2.2: Štruktúra správy SPAT.

Napríklad riadiaca jednotka pracuje len s konštantnými či dynamicky meniacimi sa dĺžkami signalizačných stavov, riadiaca jednotka je vypnutá, riadiaca jednotka detekovala hardvérovú chybu atď. Posledným povinným parametrom je stav pohybu, ktorý je popísaný v kapitole nižšie.

Voliteľné parametre stavu križovatky sú názov, minúta v roku, časová značka, povolené pruhy a asistent manévrov.

2.2.1 Stav pohybu

Každý stav pohybu (Movement State) má vlastný identifikátor, pomocou ktorého sa zviaže s elementom prepojenia pruhov v správe MAP. Vďaka tomuto prepojeniu si vie prichádzajúce vozidlo priradiť aktuálnu udalosť pohybu k prepojeniu pruhov, cez ktoré bude vozidlo prechádzať. Rovnaký princíp platí aj pre ostatné typy pruhov. Voliteľne je možné definovať asistenta manévrov pre konkrétne prepojenie pruhov.

2.2.2 Udalosti pohybu

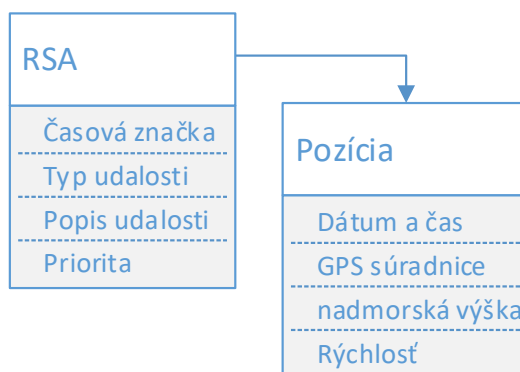
Účelom udalosti pohybu (Movement Event) je definovať v akej fáze sa aktuálny stav pohybu nachádza, napríklad fáza „zastav a zostaň stáť“ je ekvivalentom červeného svetelného signálu. Podľa stavu, v akom sa riadiaca jednotka nachádza je nutné pridať časové informácie k danej fáze. Najjednoduchší príklad je práca s konštantnými časmi, kedy je nutné špecifikovať, ako dlho bude daná fáza aktuálna.

Táto informácia je potom kľúčová pre prichádzajúce vozidlá, ktoré si z dostupných informácií vypočítajú, či je ešte možné bezpečne križovatkou prejsť, alebo v čase príchodu ku križovatke už nebude možné prejsť, pretože riadiaca jednotka bude vo fáze „zastav a zostaň stáť“.

Každý stav pohybu môže mať až šesťnásť udalostí pohybu. To znamená, že pre jedno prepojenie pruhov je možné nastaviť budúce udalosti a čas trvania fáz.

2.3 Správa pre oznamovanie hazardných situácií

Táto správa je v štandarde [3] označená stratou RSA (Road Side Alert). Z nadpisu vyplýva, že táto správa sa používa pre oznamovanie hazardných situácií a informácie v správe sa nepoužívajú pre kooperatívne predchádzanie nehodám. Typickým príkladom správy môže byť oznámenie o zľadovatej vozovke.



Obr. 2.3: Štruktúra správy RSA.

Na obrázku 2.3 je znázornená štruktúra RSA správy. Správa má len dva povinné parametre a to poradie (revízia) a typ udalosti. Poradie (revízia) má rovnakú úlohu ako v správe MAP a SPAT. Element typ udalosti je číselná hodnota kódu zo štandardizovaného slovníku viet, ktorý nesie skratku ITIS (International Traveler Information Systems). ITIS priradzuje číselné označenie ku krátkej vete, ktorá opisuje danú situáciu. Voliteľne môže správa obsahovať aj popis situácie, ktorý sa tiež skladá z ITIS

kódov. Tento slovník viet je zavedený pre zníženie dĺžky správ prenášaných systémom ITS, keďže sa prenáša len kód vety a nie jednotlivé znaky vety. Kompletný slovník viet možno nájsť v štandarde [16].

Pozícia je jeden z voliteľných elementov spravy RSA a okrem špecifikovania polohy hazardu v geografických súradniciach obsahuje časovú značku pre určenie aktuálnosti informácií obsiahnutých v správe. V prípade, ak sa zdroj hazardu pohybuje, element pozícia obsahuje aj informácie o rýchlosti a smere pohybu a ich istoty. Príkladom pohybujúceho hazardu môžu byť vozidlá používané pre správu ciest.

3 NÁVRH A IMPLEMENTÁCIA APLIKÁCIE PRE DEFINOVANIE KRIŽOVATIEK

Aplikácia má poskytnúť prostriedky, ktorými môže užívateľ zdefinovať geometriu križovatky a definovať všetky jej elementy, tak ako vyžaduje správa MAP. Ku každej definícii geometrie križovatky môže užívateľ doplniť definície stavov prepínania signálov a fáz svetelného signalizačného zariadenia križovatky. Z tejto definície budú neskôr vytvorené správy SPAT pre zvolené križovatky. Aplikácia taktiež umožňuje vytvárať všetky typy do teraz registrovaných výstrah, čo je ekvivalent správy RSA.

Jedná sa teda o softvérové vybavenie RSU jednotky, ktoré umožní emuláciu takýchto jednotiek. Emuláciou sa teda myslí, že aplikácia má byť schopná verne napodobniť funkciu jednej, či viacerých RSU jednotiek. Pomocou emulovanej RSU jednotky bude možné testovať a demonštrovať širokú škálu jej funkcionality v systémoch C-ITS.

3.1 Požiadavky na aplikáciu

Už z predošlých kapitol, ktoré podrobnejšie rozoberajú štruktúry a väzby medzi jednotlivými správami, vyplýva, že aplikácia bude pomerne rozsiahla. K rozsiahlym štruktúram správ pribudnú ešte niektoré základné požiadavky na aplikáciu:

- spustiteľná na rôznych platformách,
- jednojazyčná (anglický jazyk),
- možnosť emulácie rôznych scenárov,
- ukladania a nahrávanie vytvorených konfigurácií a scenárov,
- automatické spustenie s preddefinovanou konfiguráciou.

K týmto základným požiadavkám pribúdajú ďalšie, ktoré sú postupne popísané v nasledujúcom texte.

Aplikácia má umožniť simuláciu dvoch scenárov. V prvom scenári je funkcia emulovanej RSU jednotky totožná z klasickou, kde je jednotka umiestnená na mieste križovatky, ktorej geometriu a stavy svetelných signálov vysiela a je umožnená priama synchronizácia svetelných stavov križovatky. Druhý scenár slúži najmä pre demonštračné účely. Užívateľ má možnosť zdefinovať viacero geometrií križovatiek vo väčšej geografickej oblasti. Emulovaná RSU jednotka je súčasťou demonštračného vozidla a pri priblížení k niektorej s definovaných križovatiek začne vysielať správy práve o tejto križovatke. V tejto konfigurácii síce nebude možné emulovanú RSU priamo synchronizovať zo svetelnou signalizáciou križovatky. Synchronizáciu stavov križovatky bude možné vykonať ručne pred začatím samotnej simulácie. Oba scenáre umožňujú simulovať veľkú časť rôznych situácií, ktoré môžu nastať v systémoch

C-ITS s ohľadom na geometriu a stavy svetelnej signalizácie zvolených križovatiek.

Vzhľadom na správu MAP, aplikácia bude musieť spracovať väzby medzi zvolenou križovatkou a jej pruhmi. Musí byť možné nastaviť všetky atribúty každého pruhu, vrátane vytvorenia prepojení medzi pruhmi. Pozície križovatiek a jej pruhov musia verne kopírovať rozmiestnenie križovatky v reálnom svete. Ako predloha pre určenie týchto pozícií bude slúžiť rozhranie s Google mapou. Vzhľadom k druhému scenáru, aplikácia bude musieť umožniť definovanie viacero takýchto križovatiek.

Ďalšou zložitejšou požiadavkou je definícia správy SPAT. Táto správa okrem jej veľkosti a priamemu zaviazaniu so správou MAP, obsahuje aj dynamicky meniace sa dáta. Pri definícii položiek správy SPAT, aplikácia musí umožniť vytvorenie cyklu stavov pre svetelnú signalizáciu. Aplikácia musí umožniť tieto cykly priradiť k prepojeniam pruhov, ktoré užívateľ vytvorí pri definícii MAP správy.

Už pri definícii jednoduchšej svetelnej križovatky, pozostávajúcej z ôsmich pruhov a štyroch semaforov, správy MAP a SPAT dokopy obsahujú viac ako 200 rôznych parametrov.

V prípade správy RSA sú požiadavky jednoduchšie. Aplikácia v tomto prípade musí umožniť definíciu polohy výstražnej správy, opäť rozhranie s Google mapou. A taktiež v zrozumiteľnej podobe k jednotlivým výstrahám priradiť typ výstrah, popis a jej ďalšie parametre.

Takéto komplexné požiadavky viedli k vytvoreniu pomerne veľkému počtu softvérových modelov, ktoré spĺňajú jednotlivé požiadavky. V nasledujúcich kapitolách sú tieto bloky podrobne popísané, vrátane ich funkcionality a väzby medzi nimi.

3.2 Výber implementačných prostriedkov

Emulačná jednotka RSU, na ktorej bude aplikácia spustená je typ vstavaného počítača postaveného na architektúre ARM, ktorý je vybavený perifériami podporujúce IEEE 802.11p, GPS a Ethernet. Pripojenie k vstavanému počítaču je realizované pomocou zabezpečeného terminálu cez klasickú ethernetovú sieť. Použitý operačný systém je odľahčený typ Linux-u, ktorý disponuje ovládačmi a aplikačným rozhraním pre prácu s oboma popisovanými V2X komunikačnými modelmi.

Z uvedených požiadaviek na aplikáciu a konfigurácie cieľového systému jednotky RSU vedie na aplikáciu typu klient–server. Keďže klientská časť programu má byť spustiteľná na rôznych platformách a vývoj aplikácie pre rôzne platformy prináša veľa redundantnej práce. V snažení predísť tejto redundantnej práci sa ponúka riešenie pomocou webovej aplikácie, ktorá sa bude môcť otvoriť na každej platforme s webovým prehliadačom. Z pohľadu webových aplikácií sú moderné prehliadače výkonným implementačným nástrojom s dostatočnými prostriedkami pre realizáciu aplikácie pre definovanie geometrie križovatiek.

Pre serverovú časť programu na jednotke RSU je použitý programovací jazyk Python. Keďže jazyk Python je interpretovaný, zdrojové kódy sú jednoducho prenositeľné medzi rôznymi platformami, čo umožní väčšiu časť aplikácie napísať a odladiť na desktopovom počítači, kde bude možné použiť pokročilejšie nástroje, ktoré zjednodušia vývoj aplikácie.

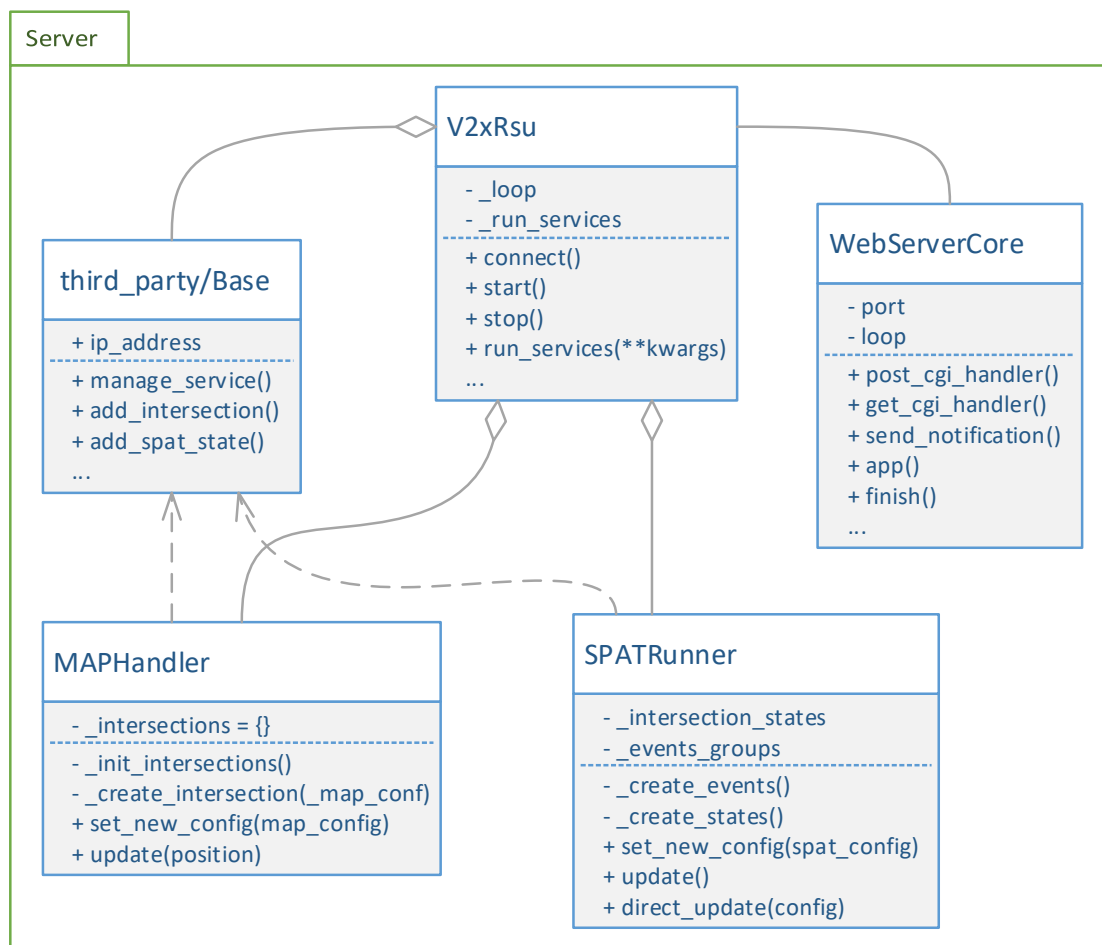
3.3 Serverová časť programu

Úlohou serverovej časti programu je vytvoriť a rozposielať správy o geometrii križovatky (MAP), stavov jej signálov (SPAT) a prípadne správy o hazardných situáciách (RSA). Obsah správ a spôsob rozposielania je obsiahnutý v konfiguračnom súbore, ktorý vygeneruje klientská časť programu. Ďalšou úlohou serverovej časti bude sprostredkovať klientsku časť programu pre užívateľa pomocou webového serveru.

Keďže serverová aplikácia musí obsluhovať niekoľko prostriedkov a spojení naraz, je výhodné aplikáciu rozdeliť do viacerých funkčných blokov. Jednotlivé bloky aplikácie sú napísané asynchrónnym spôsobom, čo zaručí, že každý z týchto blokov je možné spustiť súbežne. Pre tento typ implementácie je v jazyku Python dostupná knižnica *asyncio*, ktorá disponuje prostriedkami pre pseudo-súbežné vykonávanie blokov. Asynchrónne napísaný blok sa v knižnici *asyncio* nazýva korutina (coroutine). Pomocou knižnice *asyncio* sa vytvorí slučka pre beh programu, následne sa do nej pridajú všetky korutiny a spustí sa jej vykonávanie. V základe je potom slučka obsahujúca korutiny spustená v jednom vlákne a za prepínanie medzi korutinami zodpovedá práve spustená korutina, ktorá musí uvoľniť prostriedky¹, aby sa spustila ďalšia korutina. Jedná sa teda o spôsob jednoduchého kooperatívneho multitaskingu. Vďaka tomu, že sa programová slučka vykonáva v jednom vlákne a vždy je spustená len jedna korutina nie je nutné ošetrovať synchronizáciu medzi zdieľanými premennými v programe.

Na obrázku 3.1 možno vidieť UML diagram serverovej aplikácie a spomínané rozdelenie do funkčných blokov. Každý relevantný blok je navrhnutý tak, aby mohol byť spustený asynchrónne. V prípade, ak aplikácia nebude dostatočne rezponzívna je jednoducho možné prejsť k viacprocesorovému spracovaniu, ktoré taktiež sprostredkúva knižnica *asyncio*. Po prechode k viacprocesorovému spracovaniu je už nutné ošetrovať zdieľané prostriedky medzi procesmi. Na túto skutočnosť som pri návrhu aplikácie myslel a preto každý modul má samostatné prostriedky pre uchovávanie behových dát, takže sa nemusia zdieľať s ďalšími modulmi. Taktiež každý modul poskytuje minimálny nutný počet metód pre vstup a pre výstup.

¹Prostriedky sa zvyčajne uvoľňujú pri čakaní na I/O operáciu, alebo pri čakaní na odpoveď komunikácie.



Obr. 3.1: UML diagram serverovej časti programu.

Vďaka tomu je možné jednoducho moduly spúšťať ako samostatný proces a celkovo zvýšiť rýchlosť behu aplikácie. Rovnaká skutočnosť platí aj pre modul webového servera, ktorý môže byť spustený v samostatnom procese. Pri tomto riešení by stačilo doplniť synchronizáciu konfiguračného súboru a priamych užívateľských vstupov z klientskej aplikácie napríklad zapnutie a vypnutie vysielania V2X správ. V nasledujúcich kapitolách sú podrobnejšie popísané jednotlivé bloky aplikácie a ich zaujímavé časti. Kompletná dokumentácia zdrojového kódu serverovej časti programu je dostupná na priloženom médiu.

3.3.1 Modul infraštruktúrneho bodu

Modul infraštruktúrneho bodu je hlavným modulom celej serverovej časti aplikácie. Tento modul je reprezentovaný triedou z názvom *V2xRsu* a disponuje prostriedkami pre nadviazanie a správu pripojenia k samotnej V2X knižnici pomocou adaptéru.

Ďalšie funkcie sú:

- periodická aktualizácia polohy RSU jednotky z GPS prijímača,
- poskytovanie REST² API pre manipuláciu pomocou web servera,
- vyvolávanie notifikácií na závažné udalosti,
- spúšťanie a ukončovanie vysielania V2X správ,
- registrovanie metód pre spätné volania V2X knižnice,
- povolenie, či zakázanie rôznych služieb (MAP, SPAT, DirectSPAT, RSA,...),
- manipulácia konfiguračného súboru.

Vďaka týmto funkciám tento modul poskytuje robustné rozhranie pre správu a konfiguráciu RSU jednotky.

REST API je určené pre interakciu s webovým serverom, aktuálne umožňuje zapínať a vypínať vysielanie V2X správ, poskytuje polohu a stav RSU jednotky. Napríklad po zadaní url adresy `rsu_ip_address:port/cgi/status`, server odpovie s JSON (JavaScript Object Notation) objektom o stave RSU jednotky. Podobným spôsobom fungujú aj notifikácie, tie však používajú technológiu WebSocket. V tomto prípade webová aplikácia nadviaže spojenie na dopredu definovaný koncový bod (`/api/notifications/`) a udržiava toto spojenie. Modul infraštruktúrneho bodu potom cez toto transparentné spojenie posiela notifikačné správy jednotného formátu, na ktoré môže klientska časť reagovať okamžite po prijatí. Vďaka tomuto riešeniu môže RSU jednotka užívateľa okamžite upozorniť o akejkoľvek zmene svojho stavu, či upozorniť na rôzne ďalšie udalosti.

V2X knižnica taktiež vyvoláva spätné volania, čo môže byť reakcia na rôzne typy udalostí, ku ktorým sa aplikácia prihlási počas inicializácie modulu či počas behu programu. Obslúženie týchto volaní spočíva k zaregistrovaní korutiny práve pre obsluhu jedného volania a v prípade, že toto volanie nastane, adaptér V2X knižnice pridá konkrétnu korutinu do *asyncio* slučky, ktorá bola adaptéru priradená pri inicializácii modulu. Niektoré zo spätných volaní priamo vyvolajú notifikáciu pre užívateľa, tak ako bolo popísané v predošlom bloku. Príkladom môže byť reakcia na stratenie GPS signálu. Adaptér V2X knižnice je podrobne popísaný v kapitole 3.3.5

Pre manipuláciu s konfiguračným súborom obsahuje trieda *V2xRsu* vnorenú triedu *Scenario*. Objektu, ktorý je vytvorený z tejto triedy, sa vždy pred spustením vysielania V2X správ predá najaktuálnejšia konfigurácia. Podľa užívateľom zvoleného scenára sa vygenerujú modifikované konfigurácie, pre ostatné moduly, ako napríklad modul *SPATRunner*. Zároveň *Scenario* určí, ktoré RSU služby sa majú spustiť a ktoré nie. Služba je v tomto kontexte myslená ako funkcionálna niektorého z modulov, ktorý je schopný generovať niektorý typ V2X správy. Napríklad, ak bude služba SPAT vypnutá, jednotka RSU nevyšle žiadnu SPAT správu, ale ostatné

²REST (Representational State Transfer) je typ bezstavového komunikačného protokolu.

služby budú pracovať normálne.

Modul taktiež obsahuje inicializačnú funkciu, ktorá vygeneruje a pozbiera všetky údaje do premenných, ktoré sú potrebné pre beh programu. K tomu vytvorí *asyncio* slučku, do ktorej sú pridané všetky relevantné korutiny a spustí sa jej vykonávanie. Táto funkcia je volaná modulom CLI (Command Line Interface), ktorý slúži ako vstupný bod celého programu. Tento modul užívateľovi poskytuje ďalšie možnosti pre spustenie aplikácie pomocou niekoľkých prepínačov. Popis niektorých možností:

- *-f, -force_start* – spustí aplikáciu a začne vysielat V2X správy,
- *-c, -config_file* – špecifikuje cestu k vlastnému konfiguračnému súboru,
- *-no_web* – zakáže spustenie webového užívateľského rozhrania,
- *-p, -port* – nastaví port, na ktorom bude spustený webový server.

Kompletný výpis podporovaných parametrov možno nájsť v priloženej dokumentácii k programu.

3.3.2 Generovanie správ geometrie križovatky

Pre generovanie správ obsahujúce geometriu križovatky je vytvorený modul s názvom *MAPHandler*. Pred každým spustením vysielania V2X správ je pre *MAPHandler* pomocou metódy *set_new_config* predaná nová konfigurácia, ktorá je upravená triedou *Scenario* podľa aktuálne zvoleného scenára. Metóda *set_new_config* taktiež vyvolá vytvorenie konkrétnych objektov, ktoré už reprezentujú samotnú MAP správu, v prípade chybnjej konfigurácie metóda vyvolá výnimku.

Najdôležitejšia metóda modulu je *update*, ktorá na základe zvoleného scenára a ďalších nastavení, aktualizuje dynamické dáta, ktoré správy MAP obsahujú. Príkladom môžu byť časové značky. Spolu s dynamickými dátami, metóda *update* určí čas ďalšej aktualizácie, ktorý sa môže meniť v prípade obsluhy viacero MAP správ naraz. Metóda po každom zavolaní vráti pole objektov reprezentujúcich MAP správy a zároveň čas ďalšej aktualizácie. Metóda je volaná z korutiny, ktorá je súčasťou modulu *V2xRsu* a vrátené pole spracuje a aktualizované správy na V2X správy a začne sa ich vysielanie. Čas ďalšej aktualizácie slúži pre uspanie korutiny, čím sa uvoľnia prostriedky pre ďalšie korutiny.

Metóda má jeden vstupný parameter *position*, ktorý reprezentuje GPS súradnice RSU jednotky. Súčasťou každej MAP správy je element 3D pozície, ktorý obsahuje GPS súradnice kotvy križovatky. Tieto súradnice slúžia pre simulovanie dosahu RSU jednotky v prípade druhého scenára, kedy je jednotka RSU v rovnakom vozidle, ako jednotka OBU. Pomocou aktuálnej polohy RSU jednotky a GPS súradníc kotiev od každej križovatky v konfigurácii sa spočíta vzdialenosť. Táto vzdialenosť sa porovná so simulovanou hodnotou dosahu a RSU jednotka vygeneruje len MAP správy,

ktoré sa nachádzajú v simulovanom dosahu. Tento prístup odľahčí RSU jednotku od generovania nerelevantných MAP správ.

3.3.3 Generovanie správ pre signály a fázy križovatky

Generovanie správ pre signály a fázy križovatky obstaráva modul *SPATRunner*. V nasledujúcom texte sú pre prehľadnosť **tučným písmom** vyznačené časti SPAT správy. Tento modul je oproti predošlému násobne zložitejší, hlavne pre nutnosť aktualizácie a stráženiu množstva rôznych položiek naraz. Pre interakciu s modulom je vytvorené rovnaké rozhranie ako v prípade MAP modulu. Opäť pred každým spustením vysielania V2X správ je modulu *SPATRunner* predaná nová konfigurácia pomocou metódy *set_new_config*, ktorá pomocou ďalších privátnych metód spracuje konfiguráciu a vytvorí objekty reprezentujúce **stav križovatky** (obrázok 2.2) a skupiny udalostí (*EventGroup*), ktoré sa uložia do stromových štruktúr slovníku a setu.

Modul *SPATRunner* je navrhnutý tak, aby mohol udržiavať a aktualizovať viacero **stavov križovatiek** naraz. Pri návrhu som vychádzal zo štruktúry SPAT správy na obrázku 2.2, z ktorej vyplýva, že najviac aktualizovaných informácií je súčasťou **stavov pohybu**. V reálnom svete **stav pohybu** môže byť reprezentovaný svetelným signalizačným zariadením, napríklad semaforom a **udalosť pohybu** je stav, v ktorom sa semafor nachádza, napríklad svieti červená farba. Pri každej zmene sa musia aktualizovať potrebné **udalosti** v **stavoch pohybu** a ich časové informácie. Keďže týchto **stavov pohybu** môže byť v jednej konfigurácii pomerne veľké množstvo, vytvoril som ďalšiu podtriedu s názvom *EventGroup*.

Pomocou podtriedy *EventGroup* má modul *SPATRunner* potrebné prostriedky pre prácu zo **stavmi pohybu** a ich **udalosťami**. Hlavný účel podtriedy *EventGroup* je zhromaždiť všetky **udalosti** pre jeden **stav pohybu**, príkladom môže byť semafor číslo 1 má 2 udalosti, zastav a zostaň stáť (červená) s dĺžkou 15 sekúnd a prejazd povolený (zelená) s dĺžkou 10 sekúnd. *EventGroup* taktiež oddeľuje **stavy pohybu** od každej križovatky. Toto rozdelenie neskôr zjednoduší aktualizáciu informácií o stavoch križovatiek v metóde *update*. Pre vytvorenie objektu z *EventGroup* je nutné zadať nasledujúce parametre:

- číslo ID označujúce *EventGroup* – slúži pre zviazanie s **udalosťou pohybu**,
- číslo ID križovatky – s predošlým číslom ID tvorí nezameniteľný identifikátor,
- názov – označuje *EventGroup* pre ladenie algoritmu.

Každá pridaná **udalosť** by mala mať správne nastavený jej stav a dĺžku jeho trvania v desatinách sekúnd. Taktiež záleží na poradí, v akom sú udalosti pridané, pretože sú staticky uložené do poľa, ktoré tvorí uzavretý cyklus udalostí. Napríklad pre semafor sú to typicky štyri rôzne stavy v jednom cykle. Po pridaní všetkých stavov

je nutné zavolať metódu *sync*. Metóda má jeden vstupný parameter a to časovú značku v desatinách sekúnd, ktorá reprezentuje začiatok prvej pridanej udalosti v cykle. Časová značka je vo formáte 64bitového času v sekundách. Na základe tejto značky sa určí, ktorá udalosť je aktuálna a koľko času tejto udalosti zostáva.

Pomocou metódy s názvom *update_and_get_event* je možné získať aktuálnu udalosť z objektu *EventGroup*. Tato metóda zistí, či je potrebná aktualizácia podľa aktuálneho času a času ďalšej zmeny, ktorý je uložený ako privátna premenná triedy. V prípade, že je zmena potrebná, vykonajú sa nasledujúce kroky:

- z pola statických udalostí sa vyberie ďalšia udalosť v poradí,
- cyklicky sa inkrementuje index pola statických udalostí pre ďalšiu aktualizáciu,
- vytvorí sa nová udalosť rovnakého typu ako vybraná statická udalosť,
- z vybranej statickej udalosti sa vyberie jej dĺžka trvania,
- vytvorenej udalosti sa aktualizuje čas konca udalosti z aktuálneho času a dĺžky trvania statickej udalosti vo formáte, ktorý je požadovaný v štandarde³,
- metóda vráti *EventGroup* ID, novo vytvorenú udalosť a dĺžku trvania udalosti.

V prípade, že zmena nie je potrebná, metóda vráti rovnakú trojicu parametrov. Prvé dva sú nastavené na neplatné hodnoty a posledný je čas ďalšej aktualizácie. *EventGroup* tiež obsahuje niekoľko statických privátnych a verejných metód pre prácu s časom. Metódy slúžia na získavanie časov v rôznych formátoch, ktoré sa používajú pri generovaní nových udalostí, synchronizácií a aktualizácií udalostí.

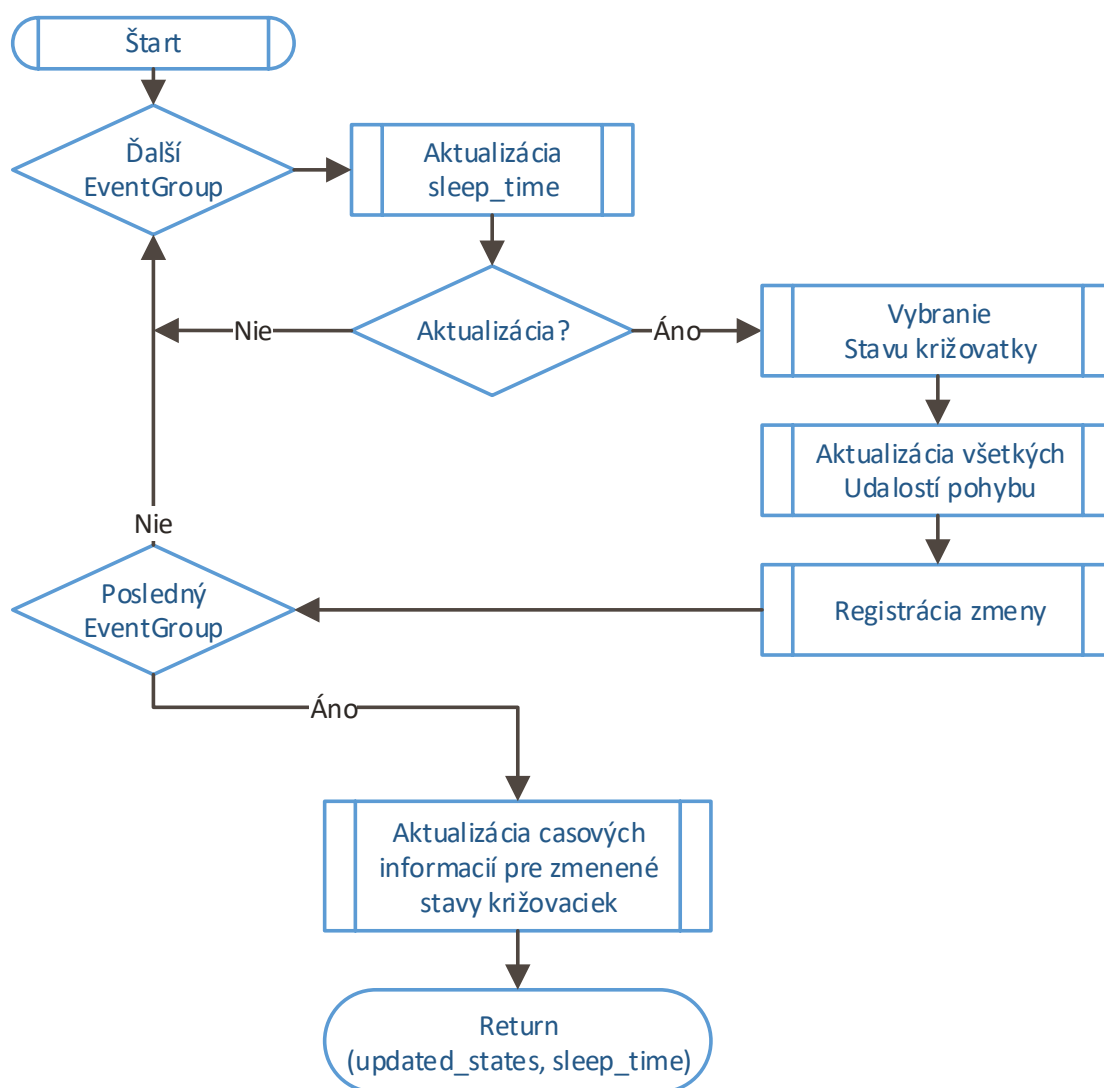
Pre aktualizáciu a vytvorenie V2X správ sú v module *SPATRunner* dostupné dve metódy, už spomínaná metóda *update* a metóda *direct_update*. Metóda *update* je kľúčová pre vytváranie SPAT správ pre každú križovatku, ktorá bola definovaná v predanej konfigurácii pomocou metódy *set_new_config*. Po zavolaní *update* metóda vykoná nasledujúce kroky:

1. v cykle prejde všetky objekty typu *EventGroup* a pre každý zavolá už predošle popísaná metóda *update_and_get_event*, ktorá vráti číslo ID *EventGroup* objektu, aktualizovanú udalosť a dĺžku trvania udalosti,
2. z dĺžky trvania udalosti sa určí minimum pre čas ďalšej zmeny,
3. ak metóda vráti platné hodnoty prejde sa k aktualizácii **stavov pohybu**,
4. podľa čísla ID križovatky uloženého v objekte *EventGroup* sa vyberie objekt **stav križovatky**,
5. zo **stavu križovatky** sa vyberú všetky **stavy pohybu**, ktoré spravuje objekt *EventGroup*,
6. pre každý vybraný **stav pohybu** sa aktualizuje predošle vrátená udalosť,
7. následne sa každý **stav pohybu** aktualizuje v **stave križovatky**,

³Udalosť používa formát času desatina sekundy v hodine, aktuálna hodina je špecifikovaná v stave križovatky.

8. zaregistruje sa číslo ID zmeneného stavu križovatky a pokračuje sa ďalej v cykle od kroku 1,
9. po prejení všetkých objektov typu *EventGroup* sa v zmenených stavoch aktualizuje časová značka,
10. následne sa vráti list **stavov križovatiek**, z ktorých sa vytvoria SPAT správy a predpokladaný čas ďalšej zmeny, ktorý sa použije pre uspanie obslužnej rutiny.

Pri aktualizácii **stavov pohybu** sa testuje, či jednotlivé stavy majú ich aktualizovanie povolené. Ak aktualizovanie **stavu pohybu** nie je povolené, znamená, že stav bol nastavený priamo pomocou metódy *direct_update*. Vývojový diagram v zjednodušenej podobe znázorňuje funkciu metódy *update* (obrázok 3.2).



Obr. 3.2: Vývojový diagram pre metódu *update*.

Metóda *direct_update* sa používa rovnako, ako metóda *update*. Jediným rozdielom je, že metóda *direct_update* umožňuje staticky nastaviť udalosť pre konkrétny **stav pohybu** pre konkrétny **stav križovatky**. To je umožnené pomocou jedného vstupného parameteru, ktorý reprezentuje konfiguráciu jednej udalosti, jej typ, dĺžku trvania, čísla ID križovatky a ID stavu pohybu. Ukážka konfigurácie je naznačená v prílohe C.1. Priamo upravovať udalosti v jednotlivých stavoch pohybu je umožnené len pre **stavy pohybu**, ktoré existujú v pôvodnej konfigurácii *SPATRunner* modulu. Toto obmedzenie vychádza zo zviazanosti SPAT a MAP správ, keďže každý **stav pohybu** vyžaduje väzbu s prepojením dvoch pruhov, ktoré je súčasťou konfigurácie MAP správy. V prípade zavolania metódy *direct_update* so správnou konfiguráciou sa aktualizuje udalosť zo zvoleným typom v **stave pohybu** a zakáže sa jeho automatická aktualizácia. Tiež je pomocou *direct_update* možné zapnúť automatickú aktualizáciu a udalosti pre konkrétny **stav pohybu** budú generované predošlým spôsobom pomocou *EventGroup*.

3.3.4 Sprostredkovanie klientskej časti programu

Klientska časť programu je realizovaná pomocou webových stránok. Pre sprostredkovanie týchto stránok má RSU jednotka implementovaný webový server, ktorý je postavený na populárnej knižnici *aiohttpd*. Funkcionalita RSU webového serveru je súčasťou modulu z názvom *webServer*. Pri vytvorení webového servera je nutné zadať cestu k pracovnému adresáru, port na ktorom budú stránky dostupné a *asyncio* slučku. Samostatný modul *webServer* má v základe len prostriedky pre manipuláciu s konfiguračnými súbormi, ktoré sa pomocou REST API rozhrania s HTTP metódami *POST* a *GET* dokážu súbory vytvárať, čítať a prepisovať. Tieto konfiguračné súbory sú zapísané v predošle zadanom pracovnom adresári.

Modul *webServer* umožňuje rozšíriť funkcionalitu REST API pomocou externe registrovateľných korutín pre obsluhu *POST* a *GET* požiadaviek. Táto rozšíriteľnosť sa využíva spolu s modulom *V2xRsu*, ktorý má pripravené práve dve korutiny pre obsluhu HTTP požiadaviek.⁴ Týmto prístupom sa dosiahne priamočiara separácia oboch modulov a ďalšou výhodou je uniformné ovládacie rozhranie *V2xRsu* modulu, ktoré je možné akýmkoľvek ďalším prístupom, napríklad ovládanie pomocou interaktívneho príkazového riadka.

Pre rôzne oznámenia o zmene stavu RSU jednotky používa modul *webServer* komunikačný protokol WebSocket. WebSocket umožňuje vytvoriť a udržiavať aktívne TCP spojenie medzi serverom a webovým prehliadačom, viac informácií možno nájsť v štandarde RFC 6455 [21]. Výhoda tohoto prepojenia je, že klientska časť aplikácie nemusí periodicky posilať požiadavky pre vyčítanie stavov RSU jednotky. RSU

⁴Funkcie rozšíreného REST API modulu *V2xRsu* boli popísané v kapitole 3.3.1.

jednotka teraz sama oznamuje zmenu stavov, ktoré sa priamo pripojeným webovým prehliadačom, klientska časť aplikácie tieto stavy spracuje a adekvátne zobrazí užívateľovi.

3.3.5 Adaptér pre V2X knižnicu

V zložke s názvom *third_party* sa nachádzajú moduly *vendor1Stack*, *dummyStack* a *base*. Tieto moduly zhromažďujú všetky potrebné prostriedky pre prácu s V2X knižnicou. Týmito prostriedkami sú jednotlivé časti V2X správ, ktoré sú reprezentované v podobe tried. Pomocou týchto tried sú vytvárané V2X správy, ako bolo popísané v predošlých kapitolách textu. Napríklad časť SPAT správy **stav pohybu** je reprezentovaná triedou *V2xMovementState*.

Modul *vendor1Stack* poskytuje základne typy objektov, pre správy MAP, SPAT a RSA, zo všetkými parametrami, tak ako boli popísané v kapitole 2. Modul realizuje rozhranie medzi statickou V2X knižnicou napísanou v programovacom jazyku C a programovacím jazykom Python. Po skompilovaní modulu *vendor1Stack* vznikne adaptér, ktorý je priamo použiteľný jazykom Python. Tento softvér je proprietárny a preto tento modul nie je súčasťou práce. Aby bola aplikácia stále spustiteľná, modul *dummyStack* plne nahradzuje rozhranie predošlého modulu. *DummyStack* taktiež umožní spustiť aplikáciu aj na rôznych architektúrach ako je cieľová ARM architektúra, alebo v prípade, keď systém nie je vybavený fyzickým IEEE 802.11p rozhraním. O tom, ktorý modul bude použitý sa stará tretí menovaný modul *base*.

Modul *base* dedí všetky základné triedy poskytnuté modulami *vendor1Stack* a *dummyStack* za účelom rozšírenia funkcionality, najmä pri indexovaní jednotlivých častí správ, ktoré sa používajú pri vytváraní V2X správ. Typický príklad je rozšírenie triedy *V2xMovementState* o vlastnosť *updatable*, ktorá je použitá pre zamedzenie automatickej aktualizácie stavov modulu *SPATRunner*, v prípade použitia priameho nastavenia stavu, ako bolo popísané v kapitole 3.3.3.

3.3.6 Synchronizácia svetelných signálov zo semaforov

RSU jednotka umožňuje synchronizovať svetelné signalizačné zariadenia križovatiek rôznymi spôsobmi. Synchronizácia svetelných stavov križovatiek je implementovaná za účelom čo najvernejšie simulovať reálne nasadenie a funkčnosť C-ITS systémov. Prvým zo spôsobov synchronizácie je pomocou časových značiek, ktorý bol popísaný v kapitole 3.3.3. Ďalším spôsobom je pomocou externých signálov, ktoré môžu priamo meniť udalosti v jednotlivých stavoch pohybu. Po spracovaní externého signálu sa konkrétna zmena aktualizuje pomocou predošle popísanej metódy

direct_update, ktorá je súčasťou modulu *SPATRunner*.

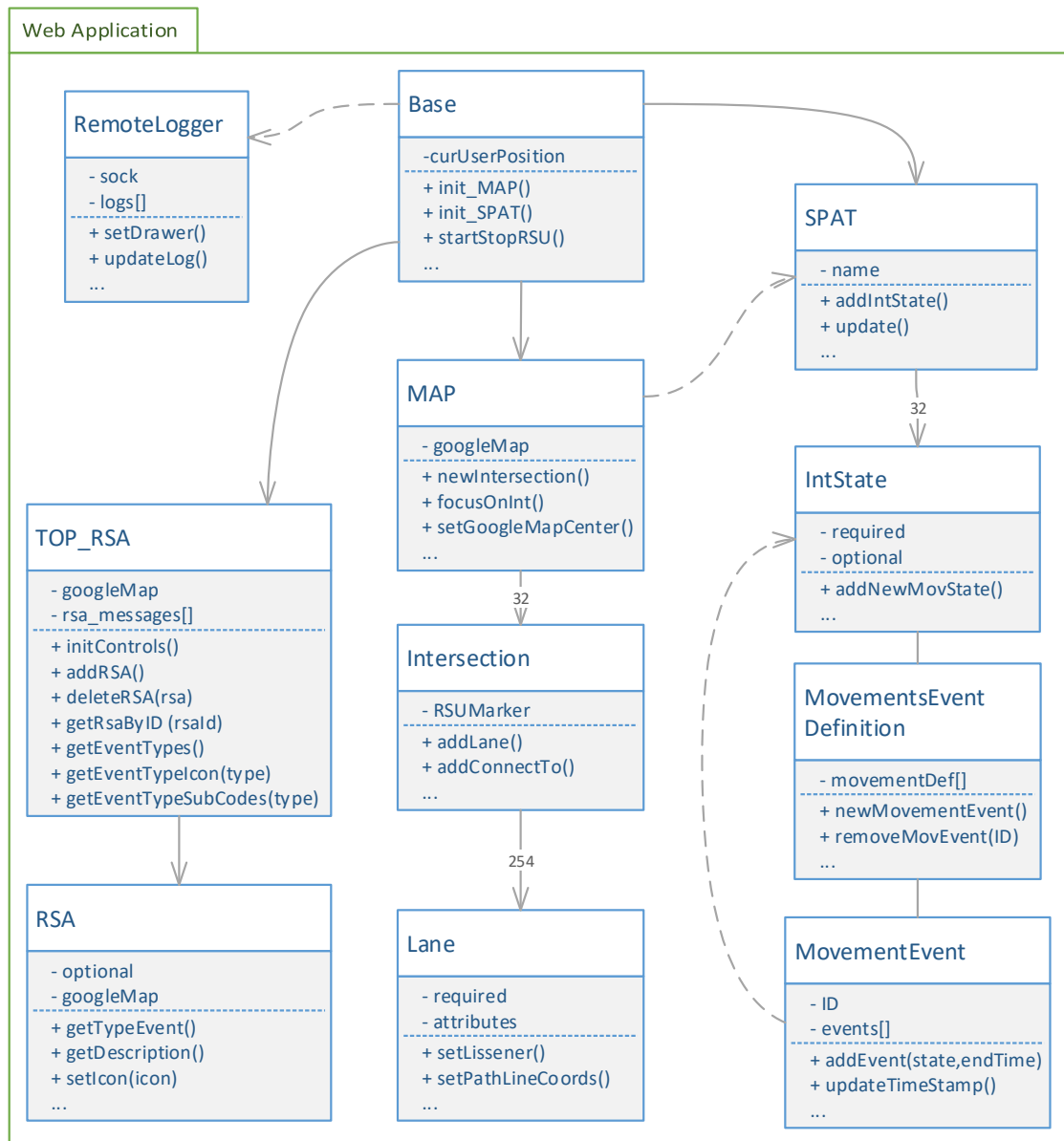
Externé signály o svetelných stavoch semaforu križovatky sprostredkuje ďalší hardvérový modul⁵, ktorý je pripojený k RSU jednotke. Tento modul pomocou sériového komunikačného rozhrania posiela informácie o stavoch jedného zo semaforov. Prijaté stavy sa synchronizujú pomocou jednoduchého algoritmu.

Synchronizačný algoritmus pracuje vo fáze učenia a fáze dozoru. Fáza učenia má za úlohu zistiť počet stavov, ich dĺžku a čas zmeny detekovaných stavov. Nasleduje detekcia cyklov v prichádzajúcich stavoch, ak sa nájdu po sebe tri cykly, s dobre prekrývajúcimi sa stavmi, algoritmus sa prepne do fázy dozoru. V stave dozoru sa vychádza z naučených časov a zmien stavov, výstup modulu sa použije len pre vyhodnocovanie istoty daného stavu. V prípade malej istoty sa algoritmus prepne opäť do fázy učenia. Výstup zo synchronizačného algoritmu je upravený pre priame použitie s metódou *direct_update*. V prípade použitia tejto synchronizácie, užívateľ v klientskej časti zvolí, ktorý stav pohybu (semafor) bude algoritmus ovládať.

⁵Modul je mikropočítač s kamerou, ktorý slúži pre rozpoznávanie svetelných signálov.

3.4 Klientská časť programu

Klientská časť programu je realizovaná pomocou webových stránok, ktoré tvoria grafické užívateľské rozhranie. Pomocou tohoto rozhrania je užívateľovi umožnené vytvoriť konfigurácie, ktoré sú serverovou časťou spracované a prevedené na konkrétne V2X správy. Štruktúra webovej aplikácie zobrazená na obrázku 3.3 odpovedá triedam, ktoré sú napísané v JavaSkripte.



Obr. 3.3: UML diagram klientskej časti programu.

Triedy zastupujú dátové štruktúry pre uvažované typy V2X správ, MAP, SPAT a RSA. Pre prehľadnosť popísané triedy udržiavajú podobnú dátovú štruktúru, ako ich

ekvivalent v DSRC správach popísaných v kapitole 2. Z týchto dátových štruktúr sú neskôr vytvorené konfiguračné súbory. Dodatočné metódy v triedach slúžia pre generovanie, upravovanie a mazanie objektov a ich vlastností na základe užívateľského vstupu. UML diagram pre prehľadnosť zobrazuje len hlavné triedy a kľúčové referencie medzi jednotlivými triedami.

3.4.1 Aplikačné rozhranie Google máp

Pri konfigurácii správy MAP z pohľadu grafického užívateľského rozhrania je kľúčové užívateľovi zobrazit geografickú mapu oblasti, v ktorej sa bude emulovaná križovatka nachádzať. Existuje veľa rôznych databáz kartografických máp s mierkami, ktoré by sa za týmto účelom dali použiť. Za jednu z týchto databáz možno pokladať mapy spoločnosti Google. Aplikačné rozhranie Google máp ponúka veľmi širokú škálu metód pre manipuláciu s mapou. Niektoré z výhod Google máp sú detailne satelitné snímky a možnosť pridať bod, ktorému odpovedá presná poloha v kartografických súradniciach. Tieto výhody ponúkajú ideálne prostredie pre definovania geometrie križovatiek. Google mapa je taktiež s výhodou použitá aj pre modul RSA, kde slúži pre umiestnenie bodov, ktoré reprezentujú polohu hazardnej situácie. Spôsob použitia je popísaný v kapitolách nižšie.

3.4.2 Hlavný modul aplikácie

Celá aplikácia je spúšťaná z modulu s názvom *main.js*, ktorý podľa aktuálne zvolenej webovej stránky inicializuje potrebné prostriedky pre vytvorenie užívateľského rozhrania. Príkladom môže byť inicializácia Google mapy pre moduly MAP a RSA pomocou funkcie *initGoogleMap*. Ďalšie funkcie, ktoré sú súčasťou modulu *main.js* slúžia pre prácu s konfiguračnými súbormi a manipuláciu s vizuálnymi časťami užívateľského rozhrania. Napríklad zakázanie ovládania po stratení spojenia so serverom.

Súčasťou modulu je trieda *Base*, ktorá inicializuje WebSocket spojenie so serverom, pomocou ktorého bude možné vytvárať upozornenia. Taktiež zistí stav v akom sa RSU jednotka nachádza a aktualizuje potrebné ovládacie prvky.

3.4.3 Modul pre vytvorenie geometrie križovatky

Tento modul sa skladá zo štvorice súborov a to *map.js*, *Intersection.js*, *Lane.js* a *formControl.js*. Každý zo súborov reprezentuje jeden celok tried a funkcií, ktoré slúžia pre vytvorenie konfigurácie pre MAP správy.

Súbor *map.js* vytvára štruktúry, ktoré umožnia interaktívnu prácu s objektom Google mapy. Pri inicializácii sa pre objekt Google mapy vygenerujú ďalšie doplnujúce ovládacie prvky, a to hlavné menu, ktoré je dostupné v ľavom hornom rohu a tiež

kontextové menu. Ak je modulu predaná valida konfigurácia, tak z nej automaticky vytvorí jej grafické reprezentácie na Google mape. V prípade definície viacerých križovatiek naraz, je pomocou metódy *focusOnIntersectionByID* umožnené medzi križovatkami prepínať.

Intersection.js je abstrakciou križovatky, ktorá je neskôr pomocou GUI zobrazená užívateľovi. Hlavná funkcionálna obsahnutá v súbore pozostáva z niekoľkých stavových automatov pre:

- interaktívne pridanie novej križovatky (*setNewRefPoint*),
- pridanie nového pruhu k aktuálne zvolenej križovatke (*addLane*),
- vytvorenie logických prepojení pruhov (*addConnectTo*, *ConnectToHandler*).

Každá križovatka je na Google mape označená bodom, ktorý reprezentuje logické umiestnenie RSU jednotky na križovatke a taktiež je ekvivalentom 3D bodu v správe MAP. Pre rýchle vyhľadávanie sa objekty pruhov patriacich ku križovatke udržujú v stromovej štruktúre pozostávajúcej z párov kľúč-hodnota. Kľúč je číslo ID pruhu a hodnota je objekt pruhu.

Lane.js reprezentuje objekt pruhu. Jeho hlavná úloha je vytvoriť zoznam bodov s kartografickými súradnicami, ktoré odpovedajú polohe, prípadne zakriveniu pruhu. Objekt pruhu taktiež umožňuje flexibilne meniť funkcie, ktoré sú volané na základe udalostí. Napríklad bežne pri kliknutí na pruh sa otvorí panel podrobných nastavení, ale v prípade vytvárania prepojení, sa po kliknutí len označí a panel nastavený sa neotvorí. Registrácia udalostí pre každý pruh je umožnená pomocou metódy *modifyListener*, ktorá sa používa najmä pri vytváraní prepojení pruhov.

Posledný súbor *formControl.js* obsahuje niekoľko funkcií najmä pre udržiavanie konzistentnosti medzi vytvorenou štruktúrou rôznych objektov a formulármi, ktoré sú užívateľovi zobrazené na webovej stránke.

3.4.4 Modul pre vytvorenie signálov a fáz križovatky

Tento modul sa skladá z troch súborov *spat.js*, *IntState.js* a *MovementsEvents.js*. Z názvov súborov možno odhadnúť ich účel. Trieda *SPAT* je opäť podobná reprezentácii správy SPAT a poskytuje základné rozhranie pre vytváranie a modifikáciu stavov križovatky. Jedna z dôležitejších metód tejto triedy je metóda *update*, ktorá zaručí konzistentnosť údajov medzi modulmi MAP a SPAT⁶.

Súčasťou súboru *spat.js* je aj trieda *DirectSPAT*, ktorá vytvára jednoduché rozhranie pre priame ovládanie konkrétneho svetelného signálu križovatky. Po zvolení niektorého z podporovaných stavov (napr. červená) sa vygeneruje požiadavka, ktorú spracuje server pomocou metódy *direct_update* v module *SPATRunner*. Aby sa zmena prejavila v hlavnom module, musí byť táto služba povolená.

⁶Naznačené prerušovanou čiarou v UML diagrame 3.3.

Vzhľadom na dynamický charakter dát obsiahnutých v správe SPAT sú definície rozdelené do dvoch funkčných celkov. Súbor *IntState.js* obsahuje všetky prostriedky pre nastavenie povolených položiek v **stave križovatky** a **stave pohybu**. Pomocou tried v súbore *MovementsEvents.js* je umožnené vytvoriť zoznam udalosti rôzneho typu, tento zoznam udalosti reprezentuje jeden cyklus svetelného signalizačného zariadenia. Tento cyklus neskôr užívateľ priradí k predošlej definícii stavu pohybu, prípadne k viacerým. Väzba medzi *MovementsEvents* a *IntState* je pomocou unikátneho čísla ID. Toto číslo spolu s číslom ID križovatky je neskôr použité pri generovaní SPAT správ.

3.4.5 Modul správ hazardných situácií

Modul pre definovanie hazardných situácií je z predošlých najjednoduchší. V tomto prípade je modul rozdelený len do dvoch súborov *top_rsa.js* a *rsa.js*. Opäť názvy súborov odpovedajú názvom tried. Objekt z triedy *top_rsa* je vytvorený hlavným modulom, ktorý predá aktuálnu konfiguráciu a inštanciu Google mapy, tak ako to bolo aj v prípade MAP modulu. Objekt *top_rsa* obsluhuje všetky objekty užívateľského rozhrania, vrátane aktualizácie zmien zvolenej RSA správy.

Objekt vytvorený z triedy *rsa* je reprezentovaný jednou značkou na mape. Vďaka tejto značke je možné určiť presnú polohu hazardnej situácie na mape. Podľa užívateľom zvoleného typu udalosti sa adekvátne zmení aj ikona udalosti. Rozlíšenie pomocou ikon slúži pre jednoduchšiu orientáciu, v prípade pridania viacero rôznych RSA správ v malej geografickej oblasti.

Typ udalosti je ekvivalentom číselného označenia kódu zo štandardizovaného slovníku ITIS [16]. Keďže sa tento slovník neustále rozširuje, je jeho ekvivalent prepísaný do súboru v JSON formáte. Tento súbor je pri inicializácii modulu načítaný objektom *top_rsa*, ktorý vygeneruje všetky aktuálne dostupné typy hazardných udalostí. Tento spôsob je zvolený preto, aby sa nemuselo zasahovať do zdrojových súborov projektu.

3.4.6 Moduly pomocných funkcií a ďalšie

Naprieč klientskou časťou aplikácie sa objavovali rovnaké požiadavky pre generovanie obsahu či prepočet hodnôt. Všetky funkcie a metódy, ktoré obsluhujú tieto požiadavky sú obsiahnuté v súbore *helper_func.js*. Príklady niektorých tried a funkcií:

- transformácie medzi kartografickým a karteziánskym súradnicovým systémom,
- získanie názvu ulice z predanej kartografickej polohy (reverzní geocoding),
- generovanie obsahu listu z JSON formátu,
- interaktívne informačné okno slúžiace ako sprievodca pri rôznych nastaveniach.

Posledná časť klientskej aplikácie je modul, pre zobrazenie logovacích informácií z RSU jednotky. Tento modul otvorí WebSocket spojenie zo serverom, na ktorom sú vysielané všetky logovacie správy, ktoré sa zobrazujú v štandardnom stríme. Zdrojové súbory tohoto modulu sa nachádzajú v zložke *utils*.

3.4.7 Serializácia a ukladanie definovaných parametrov

Definované parametre v jednotlivých moduloch sa postupne preložia do formátu JSON. Formát JSON je priamym obrazom objektov vytvorených v JavaSkripte a je dobre čitateľný, takže sa jednoduchšie hľadajú chyby pri jeho implementácii. Ďalšia výhoda použitia tohoto formátu je nezávislosť na programovacím jazyku a to tento formát predurčuje pre výmenu štrukturovaných dát.

Užívateľom nastavené parametre a prvky správ MAP, SPAT a RSA sú reprezentované v stromovej štruktúre objektov. Každý zo zložitejších objektov má implementovanú serializačnú metódu *toJSON*, ktorá sa zavolá od každého objektu. V tejto metóde sú presne vyšpecifikované, ktoré dátové položky sa z daného objektu vrátia v JSON formáte. Vo webovej aplikácii sa metóda *toJSON* zavolá z najvyššieho bodu stromovej štruktúry čo je objekt triedy *Base*. JSON reprezentácia vytvorených objektov v textovej podobe slúži ako konfiguračný súbor pre celú aplikáciu. Podľa užívateľných požiadaviek sa konfiguračný súbor môže použiť ako aktuálna konfigurácia RSU jednotky, alebo sa môže uložiť na RSU jednotku alebo si ho môže užívateľ stiahnuť do svojho zariadenia. Ukážka najvyššej vrstvy konfigurácie je zobrazená v prílohe C.2 a funkčné príklady konfigurácií sú na priloženom médiu.

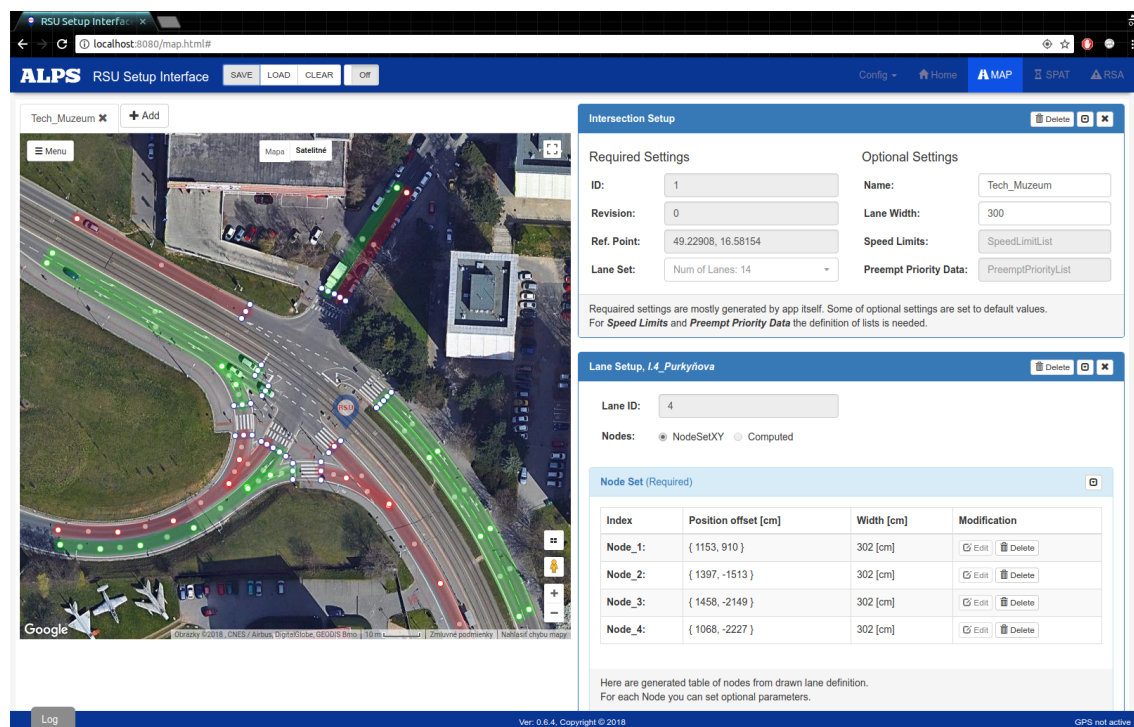
3.5 Užívateľské rozhranie programu

Vizuálna časť užívateľovho rozhrania je vytvorená pomocou štandardných webových technológií HTML5 a CSS. Pre vytvorenie jednotného vizuálneho štýlu je použitá knižnica Bootstrap 3.3.7, ktorá umožňuje konzistentné zobrazenie naprieč rôznymi rozlíšeniami zobrazovacích monitorov a displejov mobilných zariadení.

Predvolený štýl pozostáva z hlavičky a päty stránky. Hlavička obsahuje základné ovládacie prvky pre prepínanie medzi rozhraniami pre definovanie správy MAP, SPAT a RSA. Ďalšie ovládacie prvky slúžia pre manipuláciu konfiguračných súborov, aktiváciu a deaktiváciu vysielania V2X správ. V päte stránky je zobrazená verzia aplikácie, aktuálne informácie o RSU jednotke a jej kompletne logovacie informácie, ktoré sa zobrazia po rozkliknutí logovacieho okna.

3.5.1 Definovanie správ popisujúce mapu križovatky

Rozhranie slúžiace pre nakreslenie geometrie križovatky je reprezentované Google mapou, do ktorej užívateľ priamo pridá body, ktoré reprezentujú elementy križovatky v správe MAP. Postup definovania križovatky možno rozdeliť do niekoľkých krokov. Ukážka navrhnutého užívateľského rozhrania je znázornená na obrázku 3.4.



Obr. 3.4: Grafické rozhranie pre definovanie MAP správ.

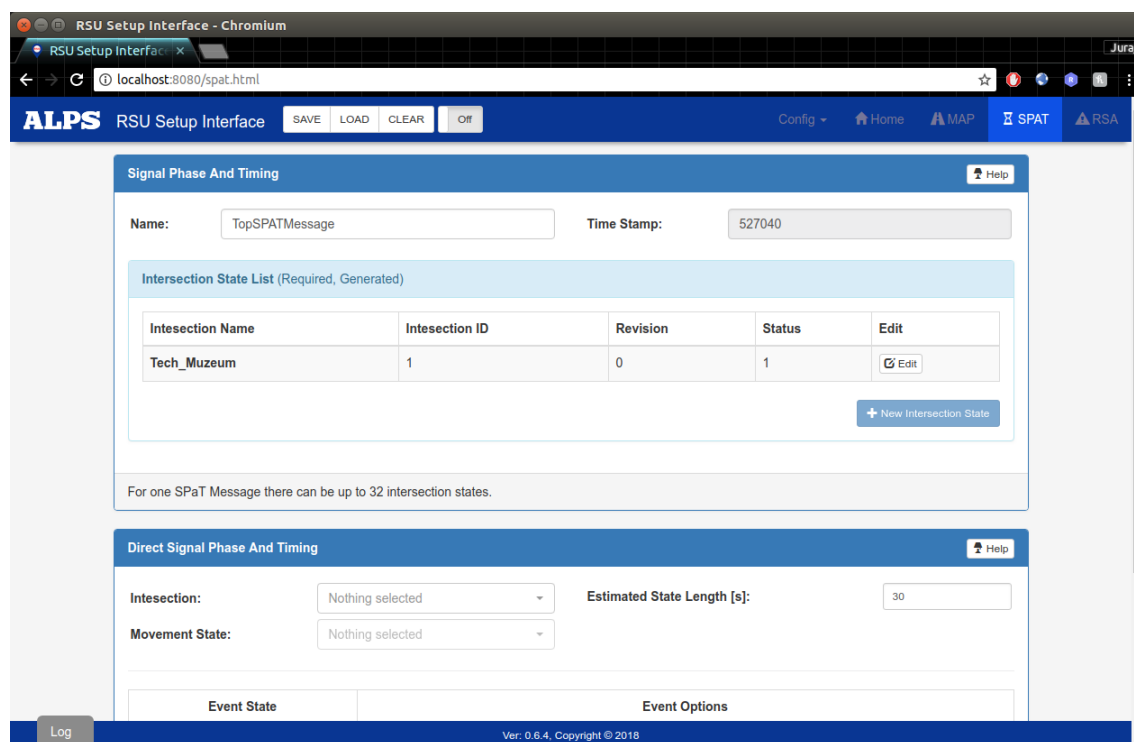
Prvý krok je nájsť križovatku, ktorú chce užívateľ definovať a pridať virtuálny bod, kde môže byť RSU jednotka umiestnená. Tento virtuálny bod reprezentuje 3D bod v správe MAP. Týmto sa vytvorí definícia jednej križovatky, ku ktorej sa v druhom kroku môžu nakresliť jednotlivé pruhy. Pre pridanie elementu pruhu, si užívateľ tento element zvolí v menu a následným klikom na mapu sa zafixuje na kliknutú polohu. Následne užívateľ upraví geometriu pruhu a jeho začiatok, tak aby čo najvernejšie kopíroval pruh zo satelitného snímku. Každý pruh je definovaný minimálne dvoma bodmi, tak ako to vyžaduje štruktúra správy MAP. V poslednom kroku môže užívateľ vytvoriť prepojenia medzi pruhmi, opäť si z menu vyberie položku pre prepojenia a pomocou kliknutí na pruhy sa automaticky prepoja. Na prepojenia sa neskôr naviažu stavy svetelnej signalizácie definované v SPAT správe.

Popisovaný postup by mal stačiť pre väčšinu jednoduchých križovatiek. Keďže táto aplikácia má byť schopná demonštrovať väčšinu výhod použitia C-ITS systémov je potrebné pokryť aj ďalšie podrobné nastavenia elementov v križovatke. Podrobné

nastavenia sa zobrazia v prehľadnej tabuľkovej štruktúre po kliknutí na jednotlivé elementy v mape. Tu môže užívateľ zmeniť niektoré automaticky vygenerované hodnoty prípadne špecifikovať ďalšie atribúty, ktoré sú súčasťou MAP správy a boli popísané v kapitole 2.1.

3.5.2 Definovanie svetelných signálov a fáz križovatky

Po kliknutí na definovanie svetelných stavov a fáz križovatiek je užívateľovi zobrazené rozhranie, ukážka rozhrania je na obrázku 3.5. Vo vrchnej časti webového rozhrania sa zobrazí vygenerovaný zoznam všetkých križovatiek, ktoré užívateľ vytvoril v predošlom kroku. Pre každú križovátku v zozname je možné otvoriť konfiguračný panel, ktorý obsahuje podrobné nastavenia **stavov pohybu** a **udalostí pohybu**. Zoznam **stavov pohybov** (*Movement States*) už bude automaticky vygenerovaný z prepojení pruhov, ktoré užívateľ vytvorí pri definovaní MAP správy.



Obr. 3.5: Grafické rozhranie pre definovanie SPAT správ.

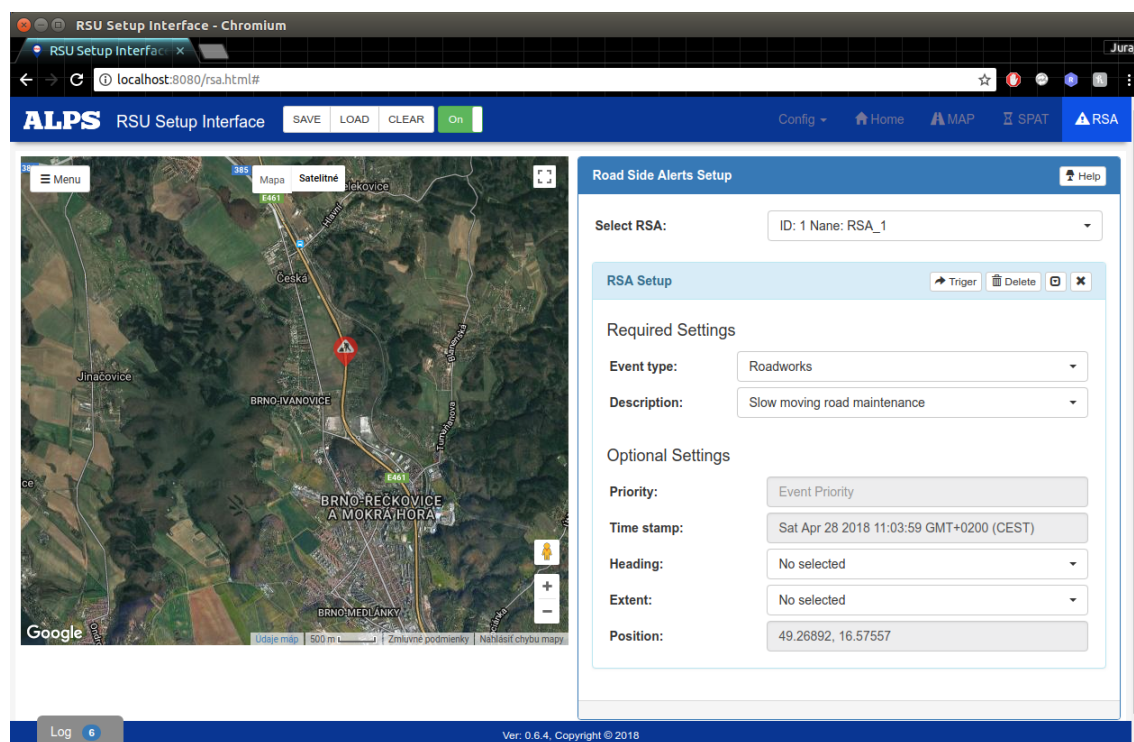
Jediný úkon, ktorý musí užívateľ v tomto štandardnom prípade vytvoriť, sú zoznamy udalostí (*Movements Event Definition*), ktoré sú ekvivalentom jedného cyklu svetelného signalizačného zariadenia a tieto zoznamy priradiť k **stavom pohybu**. V prípade, ak je k RSU jednotke pripojený hardvérový modul, ktorý poskytuje

riadenie svetelných signálov, taktiež je ho možné priradiť k niektorému z **stavov pohybov**.

Spodná časť je vyhradená pre priame manuálne nastavenie svetelných stavov, tak ako bolo popísané v predošlom texte. Ukážku tohoto rozhrania možno nájsť v prílohe B.5.

3.5.3 Definovanie hazardných situácií

Grafické rozhranie pre definovanie hazardných situácií vychádza s rovnakého usporiadania ako rozhranie pre definovanie MAP správ. Pridanie novej hazardnej situácie spočíva len z niekoľkých krokov. Prvým krokom je opäť nájsť polohu na mape, kde sa pomocou kontextového menu pridá značka. Po kliknutí na značku môže užívateľ zvoliť typ hazardnej udalosti, jej popis a ďalšie optimálne položky, tak ako boli popísané v kapitole 2.3. Ukážka konfigurácie je zobrazená na obrázku 3.6.



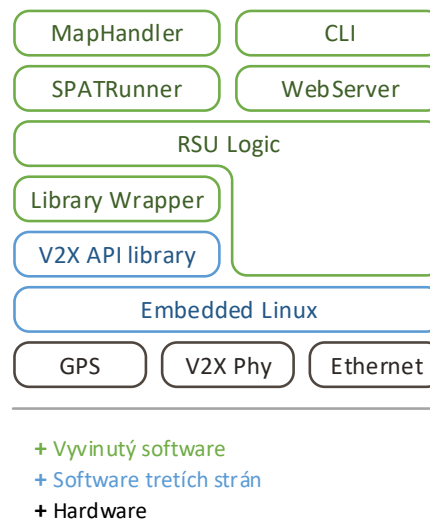
Obr. 3.6: Grafické rozhranie pre definovanie RSA správ.

4 TECHNICKÁ ŠPECIFIKÁCIA EMULAČNEJ JEDNOTKY

Vyvinutá aplikácia je určená pre referenčnú platformu firmy ALPS Electric. Referenčná platforma má nasledujúce parametre:

- V2X: IEEE 802.11p, modul ALPS UMPZ2(ES4),
- GPS: súčasť modulu ALPS UMSZ2,
- CPU: Freescale i.MX6D(ARM Cortex-A9 1GHz),
- OS: Linux Kernel 3.10.17,
- Ethernet: 10/100/1000 base,
- USB: podpora verzie 2.0,
- UART: pre Linux konzolu.

Názorné obrázky hardvéru referenčnej platformy sú súčasťou prílohy A. Na obrázku 4.1 je znázornený vrstvový model RSU jednotky. Zelenou farbou sú znázornené bloky, ktoré sú vyvinuté a popísané v tejto práci. Modrou a čiernou farbou sú naznačené časti referenčnej platformy firmy ALPS Electric.



Obr. 4.1: Stavebné bloky RSU jednotky.

Vyvinutá aplikácia, ktorá je na referenčnej platforme spustená, podporuje nasledujúci zoznam funkcií:

- emulácia rôznych scenárov,
- správa vysielania správ od rôznych modulov,
- definícia geometrie križovatiek (MAP),
- definícia stavov a fáz svetelnej signalizácie križovatky (SPAT),
- synchronizácia stavov svetelnej signalizácie križovatky (SPAT),

- manuálne ovládanie stavov svetelnej signalizácie križovatky (SPAT),
- definícia výstražných správ (RSA),
- správa vytvorených konfigurácií,
- signalizácia stavu RSU jednotky.

Pre vyskúšanie aplikácie sa na priloženom médiu nachádza obraz systému Ubuntu, v ktorom je aplikácia nainštalovaná so všetkými závislosťami. Keďže virtuálny systém nemá podporu V2X technológií, funkčnosť aplikácie nezahrňuje prácu s reálnou V2X knižnicou, tak ako bolo popísané v kapitole 3.3.5. Po spustení systému sa aplikácia automaticky spustí s jednoduchou konfiguráciou. Systémový obraz je vytvorený pomocou virtualizačného nástroja VirtualBox, verzia 5.2.

Aplikáciu je možné nainštalovať pomocou štandardných distribučných techník používaných pre šírenie Python aplikácií. Minimálne požiadavky pre spustenie aplikácie sú operačný systém typu Linux, akejkoľvek distribúcie a interpret jazyka Python verzie 3. Podrobný postup inštalácie je popísaný v dokumentácii k aplikácii na strane 1, ktorá je súčasťou priloženého média. Po úspešnom nainštalovaní a spustení aplikácie na lokálnom počítači, bez žiadnych ďalších vstupných parametrov je užívateľské rozhranie dostupné na adrese `localhost:8080`.

5 TESTOVANIE VYVINUTEJ APLIKÁCIE

V tejto kapitole sú popísané výsledky testov a postupy, ktoré boli použité pri vývoji aplikácie pre verifikáciu správnej funkcionality aplikácie. K testovaniu vyvinutej aplikácie je možno pristupovať viacerými spôsobmi. Prvým zo spôsobov je pomocou jednotkových testov, ktoré spúšťajú časti kódu a overujú jeho správne chovanie. Napríklad aproximácia kartografický súradnicový systém na karteziánsky systém. Ďalší spôsob testu je overenie správneho obsahu vysielaných správ pomocou nástroja *Wireshark*. Posledný spôsob testovania je v reálnej premávke, kde vozidlá vybavené OBU jednotkou si na základe vysielaných správ správne zobrazia potrebné upozornenia. Príkladom môže byť optimalizácia rýchlosti pre prejazd na zelenú a podobne. Pre všetky testy bola použitá referenčná platforma firmy ALPS Electric popísaná v predošlej kapitole.

5.1 Automatické testovanie zdrojového kódu

Pre jednotkové testy som zvolil *py.test*, pre flexibilitu pri písaní testov a existencie rôznych rozšírení, ktoré umožnia testovať časovo závislé metódy, načítateľné testovacie dáta a napodobňovať TCP spojenia. Jednotkové testy slúžia primárne pre otestovanie rozsahov, typov vstupných premenných a podobne.

Samotné testy sú pripravené v zložke *tests* v hlavnom adresári projektu. Testy sú rozdelené do súborov, ktoré odpovedajú modulom, ktoré testujú, napríklad modul *SPATRunner* má odpovedajúce testy v súbore s názvom *test_SPATRunner*. Aplikácia bola ladená až do doby, kedy boli všetky testy úspešné. Výsledky testov sú zobrazené vo výpise 5.1. Celkový počet testov je 54. Za každým testovacím súborom v hornej časti výpisu je niekoľko bodiek, každá bodka značí úspešný výsledok testu. V prípade, ak je test neúspešný, tak sa na mieste bodky zobrazí písmeno „F“. Dolná časť výpisu zobrazuje akú časť súborov pokryli predošle vykonané testy.

Pre minimalizáciu úkonov pred vydaním novej verzie aplikácie som použil CI (Continuous Integrations) systém. CI systém je sada nástrojov, ktoré umožnia automaticky spúšťať preddefinované úlohy. Samotné úlohy, ich funkcie a závislosti sú definované v súbore s názvom *.gitlab-ci.yml* v hlavnom adresári projektu. V súčasnosti sa úlohy pre preloženie a otestovanie aplikácie spustia po označení novej verzie aplikácie. Preklad a testy sú pripravené pre podporované architektúry ARM a x86.

CI systém umožňuje preddefinované úlohy spúšťať na viacerých platformách s rôznym hardvérovým a softvérovým vybavením. Pre zaistenie uniformného softvérového prostredia som vytvoril dva *Docker*¹ kontajnery, ktoré obsahujú všetky

¹Docker pomocou kontajnerov realizuje virtualizáciu na úrovni operačného systému, viac informácií možno nájsť na www.docker.com.

potrebné nástroje a knižnice pre preloženie a otestovanie aplikácie. Kontajner, ktorý je určený pre testovanie na ARM architektúre je ešte rozšírený o nástroje pre kompiláciu adaptéru proprietárnej V2X knižnice. Po úspešne skončených testoch sa vytvorí html verzia dokumentácie aplikácie, jej súčasťou je aj dokumentácia zdrojového kódu.

Výpis 5.1: Výsledky jednotkových testov.

===== test session starts =====			
platform linux -- Python 3.5.2, pytest-3.4.0, py-1.5.2, rootdir: /home/rsu/V2X-RSU, inifile: collected 54 items			
tests/test_MapHandler.py ...			[5%]
tests/test_SPATRunner.py			[48%]
tests/test_v2x_rsu.py			[55%]
tests/test_v2x_rsu_scenario.py ...			[61%]
tests/third_party/test_base.py			[90%]
tests/third_party/test_stack/test_dummyStack.py			[100%]
----- coverage: platform linux, python 3.5.2 - final -0 -----			
Name	Stmts	Miss	Cover
MapHandler.py	73	2	97%
SPATRunner.py	282	11	96%
__init__.py	11	0	100%
cli.py	24	9	62%
third_party/__init__.py	4	0	100%
third_party/base.py	353	59	83%
third_party/dummyStack/__init__.py	5	0	100%
third_party/dummyStack/v2xMAPWrapper.py	90	20	78%
third_party/dummyStack/v2xSPATWrapper.py	140	31	78%
third_party/dummyStack/v2xSessionWrapper.py	47	12	74%
third_party/dummyStack/v2xSimpleLog.py	57	7	88%
third_party/vendor1Stack/__init__.py	3	0	100%
v2x_rsu.py	416	212	49%
webServer.py	139	101	27%
TOTAL	1644	464	72%
===== 54 passed in 5.89 seconds =====			

5.2 Verifikácia vysielaných správ

Pre verifikáciu obsahu vysielaných správ je v hlavnom adresári projektu pripravený skript *test_static_tx.py*, ktorý po spustení na RSU jednotke, vytvorí a začne vysielat MAP a SPAT správy so statickým obsahom. Vysielané správy sú následne zachytávané na OBU jednotke, priamo z fyzického rozhrania pomocou programu *tcpdump*. Zachytené správy sú zobrazené pomocou programu *WireShark*, ktorý umožní dekodovanie obsahu správ. Dekódovaný obsah je porovnaný s pôvodným obsahom generovaním z *test_static_tx.py* skriptu. Vzhľadom na počet porovnávaných parametrov, sú výsledky testu priložené na elektronickom médiu. Test prebehol úspešne.

5.3 Testovanie funkcionality aplikácie

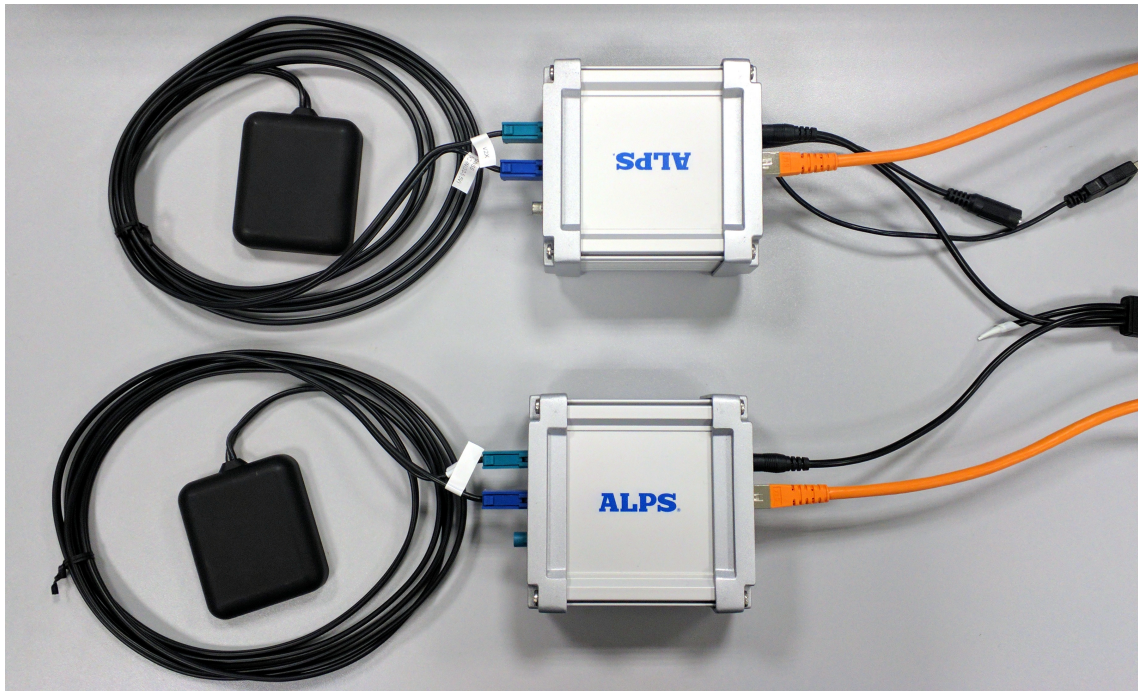
Vďaka predošlým testom je zaistené, že jednotlivé dátové položky v správach nadobúdajú správne formáty, rozsahy a hodnoty. Účel nasledujúcich testov je overiť, či štruktúrované dáta v správach dávajú zmysel a prijímajúce OBU jednotky ich dokážu správne interpretovať. Tieto testy boli vykonané v laboratórnom prostredí, kde boli všetky externé signály simulované, napríklad GPS poloha. Po úspešnom otestovaní v laboratóriu, boli RSU a OBU jednotky nainštalované v testovacom vozidle a všetky testy boli zopakované.

5.3.1 Testovanie v laboratóriu

V laboratóriu pre simuláciu polohy vozidla bol použitý nástroj *fakegps*, ktorý umožňuje prehrávať GPS záznamy v rôznych štandardných formátoch. Tieto GPS dáta sa pomocou Ethernetu, alebo sériového rozhrania predajú GPS démonovi, ktorý je spustený na OBU jednotke. Týmto testom sa má overiť správne vyvolanie výstražných správ na strane OBU jednotky. Všetky z nasledujúcich upozornení sa zobrazujú pri príchode ku križovatke:

- prejazd na červenú – RLV (Red Light Violation), sa zobrazí v prípade, že na semafore svieti červená, a zobrazí vzdialenosť ku koncu pruhu,
- čas do zeleného svetla – TTG (Time To Green Light),
- odporúčaná rýchlosť vozidla – GLOSA (Green Light Optimized Speed Advise), sa zobrazí v prípade, ak ešte stále svieti červená, a OBU jednotka si na základe dát RLV a TTG vypočíta optimálnu rýchlosť, ktorou v ideálnom prípade plynule prejde cez križovatku.

Obrázok 5.1 zobrazuje fyzickú konfiguráciu oboch jednotiek, pre testovanie v laboratóriu.



Obr. 5.1: RSU a OBU jednotky použité pre testovanie.

Tabuľka 5.1 zobrazuje výsledky testov. V tabuľke sú porovnané predpokladané hodnoty, ktoré boli určené jednoduchým výpočtom a porovnané s hodnotami, ktoré boli určené OBU jednotkou. Test predpokladá, že vozidlo sa pohybuje rýchlosťou 10 km/h.

Tab. 5.1: Výsledky testov výstražných správ.

Test	Vypočítaná hodnota	Hodnota zobrazená na OBU	Výsledok
RLV	248,12 m	248,11 m	OK
TTG	32,4 s	32,39 s	OK
GLOSA	27.57 km/h	27,57 km/h	OK

5.3.2 Testovanie v pilotnej prevádzke

V reálnej prevádzke boli testy z predošlej kapitole úspešne zopakované. Obe V2X jednotky sú do testovacieho vozidla nainštalované spolu so záložným zdrojom, ktorý zaistí konzistentné napájanie pri štartovaní vozidla. Obe jednotky taktiež prijímajú reálny GPS signál. K OBU jednotke sa pripojí dekóder CAN (Controller Area Network) zbernice pre získavanie informácií o stave testovacieho vozidla, napríklad

odometrické pulzy, ktoré slúžia pre odhad trajektórie vozidla v prípade straty GPS signálu (prejazd tunelom).



Obr. 5.2: Zoradené testovacie vozidlá (prevzaté z[15]).

S takto pripraveným vozidlom som sa zúčastnil akcie s názvom „InterCor PKI Security TESTFEST“, ktorá sa konala od 23. 4. 2018 do 26. 4. 2018 v meste Remeš, Francúzsko. Počas jedného dňa som pomocou aplikácie vysielal informácie o dvoch lokálnych svetelných križovatkách, ktoré boli na trase testovacieho okruhu, po ktorom jazdili vozidlá ostatných účastníkov (viz. 5.2). Účastníci potvrdili správne prijatie MAP a SPAT správ. Opäť zachytené správy a konfigurácie pre RSU jednotku, ktoré boli používané na akcii sú priložené na elektronickom médiu. Viac informácií možno nájsť na stránkach projektu InterCor [15].

6 ZÁVER

Práca sa zaoberá rozborom dvoch komunikačných modelov, ktoré sa používajú pre prenos správ v kooperatívnych systémoch inteligentnej dopravy (C-ITS). Jeden s modelov je určený pre Európu a druhý pre Ameriku. Oba komunikačné modely stavajú na rovnakej fyzickej vrstve definovanej štandardom IEEE 802.11p, ktorý je odvodený zo štandardu IEEE 802.11a. Pre komunikáciu je vyhradených sedem kanálov, ktoré sa rôzne využívajú v oboch modeloch. Rozdielne sú mechanizmy pre kontrolu preťaženia. Hlavný rozdiel je na sieťovej a transportnej vrstve, kde Americký model používa protokol krátkych správ (WSMP) a Európsky model používa GeoNetworking spolu s jednoduchým transportným protokolom (BTP). Protokoly na vyšších vrstvách sú medzi sebou podobné, líšia sa len v ich implementáciách vzhľadom na nižšie vrstvy komunikačných modelov.

Hlavným cieľom tejto práce bolo vytvorenie testovacej aplikácie, ktorá umožní užívateľovi jednoduchým spôsobom vytvoriť komplexné správy obsahujúce informácie o geometrii a stavoch svetelnej signalizácie križovatky pre kooperatívny systém inteligentnej dopravy. Aplikácia taktiež umožňuje definovať rôzne výstražné správy, napríklad o stave vozovky. Požiadavky na aplikáciu viedli k riešeniu pomocou aplikácie typu klient–server. Serverová časť je spustená na infraštruktúrnej jednotke RSU, ktorá vytvára a automaticky aktualizuje dynamické informácie v správach, ktoré sa vysielajú pre okoloidúce vozidlá. Tieto správy sa vytvoria z konfiguračného súboru, ktorý vygeneruje klientska časť programu. Klientska časť je realizovaná pomocou webových stránok, kde sa stavia na širokej funkcionalite moderných prehliadačov. Definovanie geometrie križovatiek je užívateľovi sprístupnené za pomoci aplikačného rozhrania Google máp. Pomocou tohoto rozhrania je možné definovať viacero križovatiek. Z definovanej geometrie križovatky aplikácia automaticky vygeneruje veľkú časť štruktúr pre správu, ktorá obsahuje informácie o fázach a stavoch svetelnej signalizácie križovatky. Užívateľ následne musí len priradiť „semafore“ k jednotlivým prepojeniam pruhov. Grafické rozhranie taktiež zobrazuje aktuálne informácie o RSU jednotke (napríklad stav GPS signálu, atď.). Architektúra oboch častí programu je navrhnutá s ohľadom na jednoduchú rozšíriteľnosť o ďalšiu funkcionalitu RSU jednotky.

Aplikácia bola rozsiahlo otestovaná rôznymi metódami, aby spĺňala všetky zadané požiadavky. Rozsiahle testovanie aplikácie zaisťuje stabilitu a spoľahlivosť behu aplikácie, zaručuje správnosť vysielaných dát.

Aplikácia je prevádzkovaná na testovacej infraštruktúrnej jednotke firmy ALPS Electric. V súčasnej dobe je používaná ako demonštrátor a prebiehajú diskusie o jej použití v pilotných projektoch v oblasti C-ITS v Českej Republike. Stabilná verzia aplikácie bola úspešne použitá pre emulovanie dvoch križovatiek počas akcie

„InterCor PKI Security TESTFEST“, ktorá sa konala v máji 2018, Remeš, Francúzsko.

Do budúcnosti sa aplikácia plánuje rozšíriť o modul, ktorý umožní definovať DENM správy, ako aj ďalšie časti systému C-ITS.

LITERATÚRA

- [1] Ahmed, S. A.; Ariffin, S. H.; Faisal, N.: Overview of wireless access in vehicular environment (WAVE) protocols and standards. *Indian Journal of Science and Technology*, rok 6. 7. 2013: s. 4994–5001.
- [2] C-ROADS *THE PLATFORM OF HARMONISED C-ITS DEPLOYMENT IN EUROPE* [online]. [cit. 23.11.2017]. Dostupné z URL:<<https://www.c-roads.eu/>>.
- [3] Dedicated Short Range Communications (DSRC) Message Set Dictionary. Surface Vehicle Standard J2735, SAE International, 2016
- [4] European Truck Platooning Challenge *What is Truck Platooning?* [online]. [cit. 31.11.2017]. Dostupné z URL:<<https://www.eutruckplatooning.com/About/>>.
- [5] ETSI EN 302 636-5-1 (V2.1.1) *Intelligent Transport Systems (ITS); Vehicular Communications; GeoNetworking; Part 5: Transport Protocols; Sub-part 1: Basic Transport Protocol*. August 2017.
- [6] ETSI EN 302 665 (V1.1.1) *Intelligent Transport Systems (ITS); Communications Architecture*. September 2013.
- [7] ETSI ES 202 663 (V1.2.1) *Intelligent Transport Systems (ITS); European profile standard for the physical and medium access control layer of Intelligent Transport Systems operating in the 5 GHz frequency band*. Júl 2013.
- [8] ETSI EN 302 636-4-1 (V1.3.1) *Intelligent Transport Systems (ITS); Vehicular Communications; GeoNetworking; Part 4: Geographical addressing and forwarding for point-to-point and point-to-multipoint communications; Sub-part 1: Media-Independent Functionality*. August 2017.
- [9] ETSI TS 102 724 (V1.1.1) *Intelligent Transport Systems (ITS); Harmonized Channel Specifications for Intelligent Transport Systems operating in the 5 GHz frequency band*. Október 2012.
- [10] Festag, A.: Cooperative intelligent transport systems standards in Europe. *IEEE communications magazine*, rok 52. 12. 2014: s. 166–172.
- [11] IEEE Std. 1609.1-2006. *IEEE Trial-Use Standard for Wireless Access in Vehicular Environments (WAVE), Resource Manager*.

- [12] IEEE Std. 802.11p-2010. *IEEE Standard for Information technology—Telecommunications and information exchange between systems—Specific Requirements—Part 11: Wireless AN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Amendment 6: Wireless Access in Vehicular Environments*.
- [13] IEEE Std. 1609.4-2016. *IEEE Standard for Wireless Access in Vehicular Environments (WAVE)—Multi-Channel Operation*
- [14] IEEE Std. 1609.0-2013. *IEEE Guide for Wireless Access in Vehicular Environments (WAVE) Architecture*
- [15] InterCor *Interoperable Corridors deploying cooperative intelligent transport systems* [online]. [cit. 29.11.2017]. Dostupné z URL:<<http://intercor-project.eu/>>.
- [16] ITIS Phrase Lists (International Traveler Information Systems). Surface Vehicle Standard J2540/2, SAE International, 2009
- [17] Liang, Z.; Wakahara, Y.: City traffic prediction based on real-time traffic information for intelligent transport systems. V *ITS Telecommunications (ITST), 2013 13th International Conference*, IEEE, 2013, s. 378–383.
- [18] NYC Connected Vehicle Project *For Safer Transportation* [online]. [cit. 2.4.2018]. Dostupné z URL:<<https://www.cvp.nyc/>>.
- [19] NGUYEN, Tien Viet, et al.: A comparison of cellular vehicle-to-everything and dedicated short range communication. V *Vehicular Networking Conference (VNC)*, IEEE, 2017. s. 101-108.
- [20] 3GPP, The Mobile Broadband Standard. *Release 14* [online]. [cit. 23.04.2018]. Dostupné z URL:<<http://www.3gpp.org/release-14>>.
- [21] RFC 6455. *The WebSocket Protocol* [online]. [cit. 23.03.2018]. Dostupné z URL:<<https://tools.ietf.org/html/rfc6455>>.
- [22] Sjoberg, K.; Andres, P.; Buburuzan, T.; aj.: Cooperative Intelligent Transport Systems in Europe: Current Deployment Status and Outlook. *IEEE Vehicular Technology Magazine*, rok 12.2.2017: s. 89–97.
- [23] US Department of Transportation. *Intelligent Transportation Systems - Connected Vehicle Pilot Deployment Program* [online]. [cit. 23.11.2017]. Dostupné z URL:<<https://www.its.dot.gov/pilots/>>.

ZOZNAM SYMBOLOV, VELIČÍN A SKRATIEK

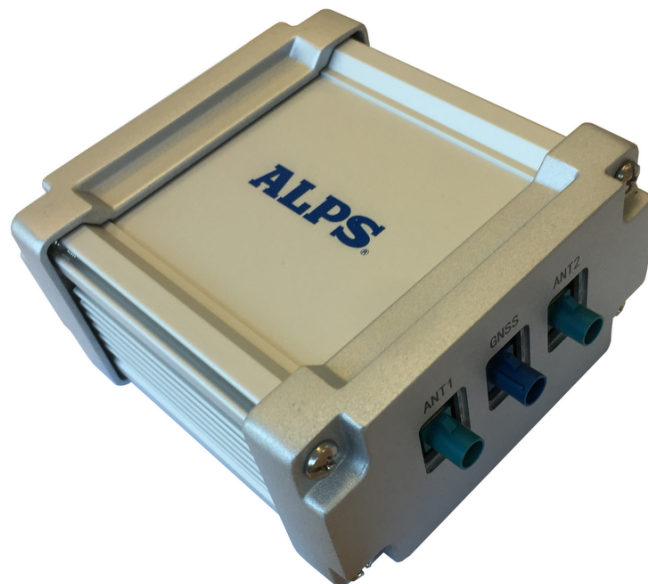
BSM	Basic Safety Message – správa o bezpečnosti
BTP	Basic Transport Protocol – základný transportný protokol
C2C	Car to Car – auto k autu
C-ITS	Cooperative Intelligent Transport System – kooperatívny systém inteligentnej dopravy
C-V2X	Cellular V2X – bunkové V2X
CAM	Cooperative Awareness Messages – kooperačné správy o informovanosti
CLI	Command Line Interface – príkazový riadok
CSR	Common Safety Request – spoločná požiadavka na bezpečnosť
DCC	Decentralized Congestion Control – decentralizovaná kontrola preťaženia
DENM	Decentralized Environmental Notification Message – decentralizované hlásenie o environmentálnom upozornení
DSRC	Dedicated Short Range Communications – vyhradené komunikácie s krátkym dosahom
eCall	Emergency Call – tiesňové volanie
EDCA	Enhanced Distributed Channel Access – rozšírený prístup k distribuovanému kanálu
ETSI	European Telecommunications Standards Institute – Európsky ústav pre telekomunikačné normy
GLOSA	Green Light Optimized Speed Advise – odporúčaná rýchlosť vozidla
ICA	Intersection Collision Avoidance – vyvarovanie kolízií na križovatkách
ITIS	International Traveler Information Systems – medzinárodné informačné systémy pre cestujúcich
ITS	Intelligent Transport System – systém inteligentnej dopravy
JSON	JavaScript Object Notation – JavaScriptový objektový zápis
MAP	Map Data – Mapové údaje
OBU	On Board Unit – palubná jednotka
PSID	Provider Service Identifier – identifikátor služby poskytovateľa
PVD	Probe Vehicle Data – dátové údaje o vozidle
RLV	Red Light Violation – prejazd na červenú
RM	Resource Manager – manažér prostriedkov
RSA	Road Side Alert – cestné upozornenie
REST	Representational State Transfer – Reprezentatívny prevod stavov
RSU	Road Side Unit – infraštruktúrna jednotka
SPAT	Signal Phase And Timing – signály fázy a časovanie
TIM	Traveler Information Message – informačná správa pre cestujúcich

TTG	Time To Green Light – čas do zeleného svetla
V2I	Vehicle to Infrastructure – vozidlo k infraštruktúre
V2P	Vehicle to Pedestrian – vozidlo k chodcovi
V2V	Vehicle to Vehicle – vozidlo k vozidlu
V2X	Vehicle to Everithing – vozidlo ku všetkému
WAVE	Wireless Access in Vehicular Environments – bezdrôtový prístup v automobilovom prostredí
WSMP	WAVE Short Message Protocol – WAVE protokol krátkych správ

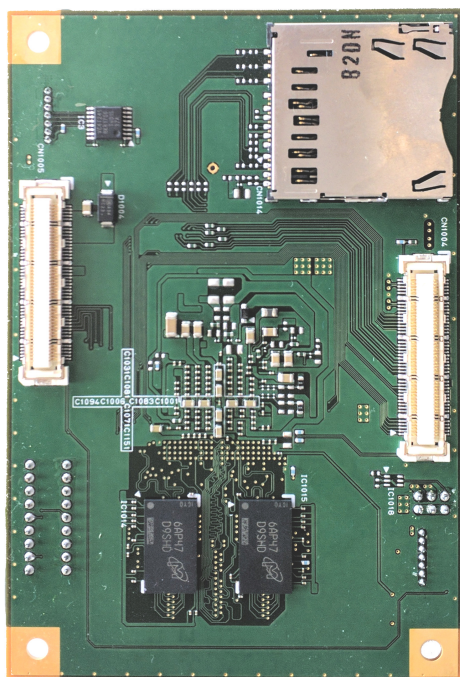
ZOZNAM PRÍLOH

A	Infraštruktúrna jednotka	68
B	Užívateľské rozhranie	70
C	Príklady konfigurácií	73
D	Obsah priloženého CD	76

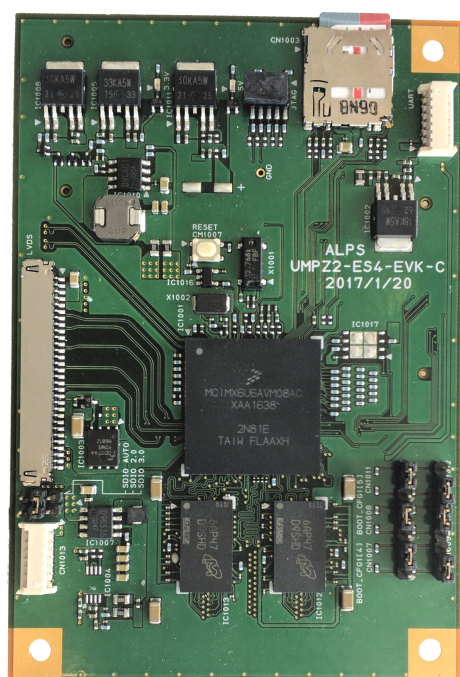
A INFRAŠTRUKTÚRNA JEDNOTKA



Obr. A.1: Infraštruktúrna jednotka RSU.

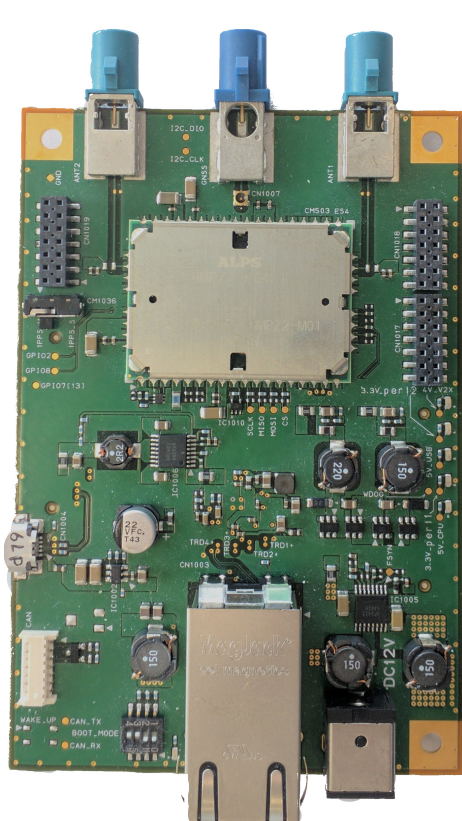


(a) Vrchná vrstva.

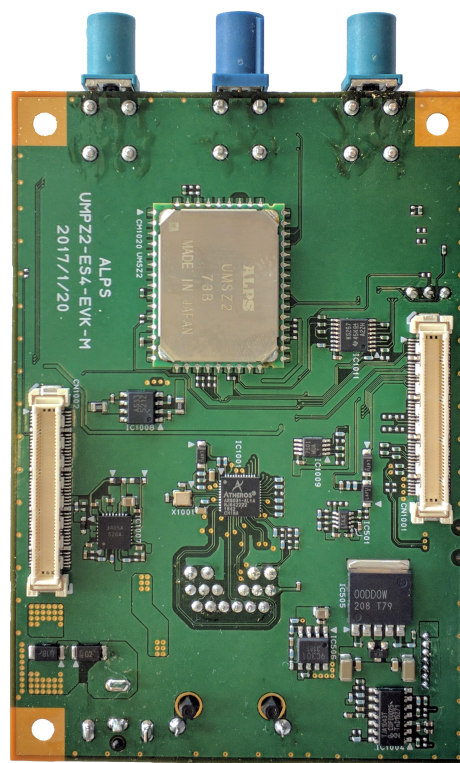


(b) Spodná vrstva.

Obr. A.2: Procesorová časť RSU jednotky.



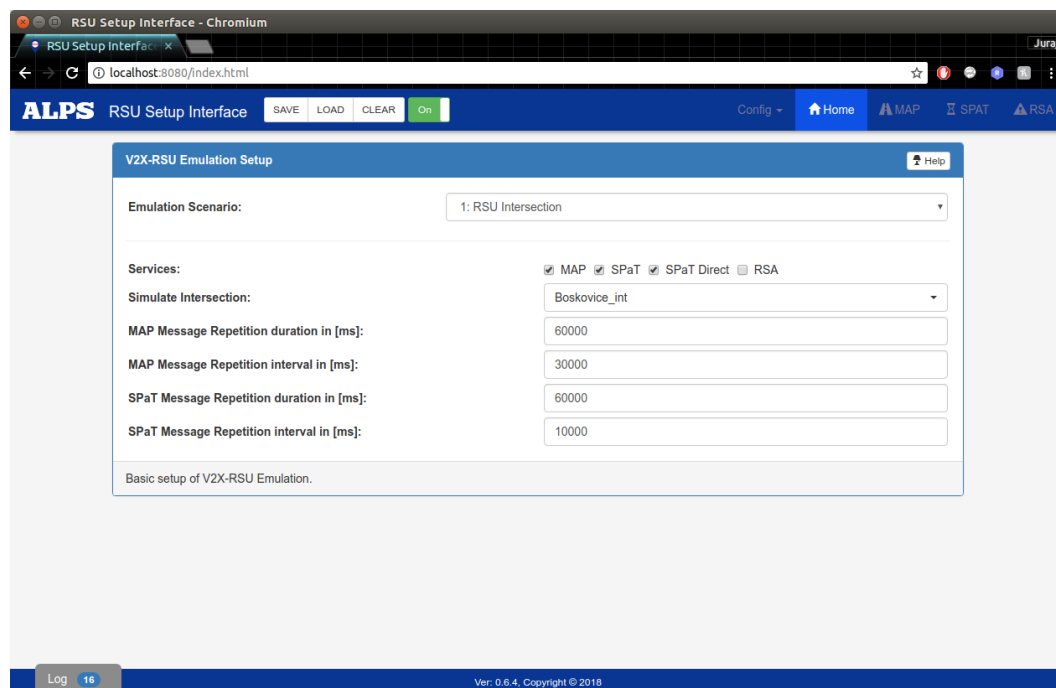
(a) Vrchná vrstva.



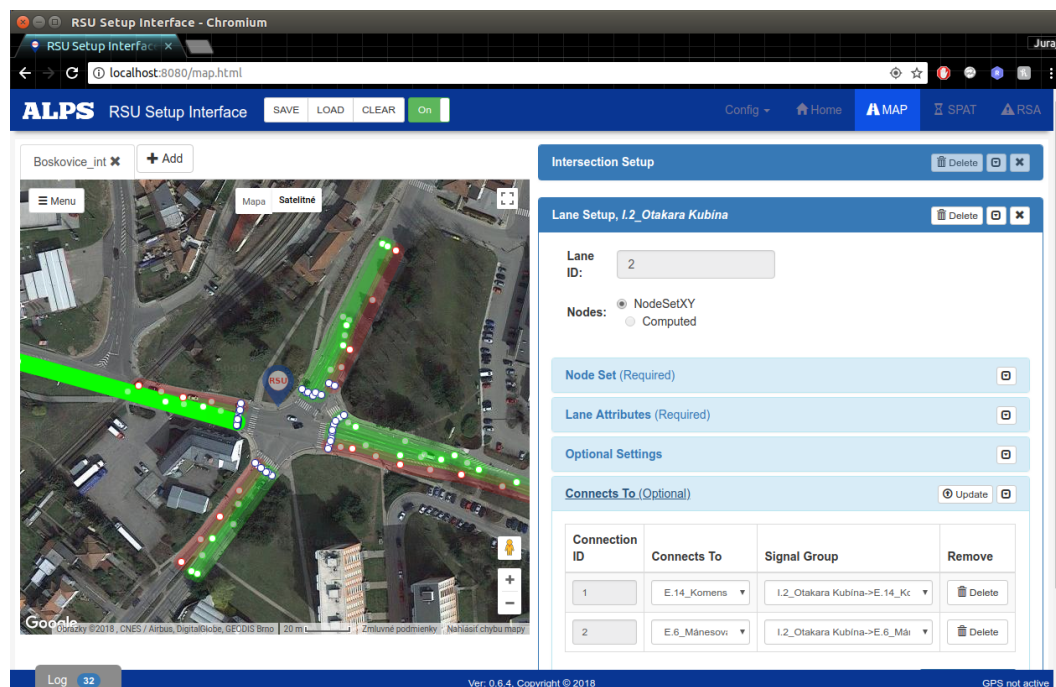
(b) Spodná vrstva.

Obr. A.3: RF časť RSU jednotky.

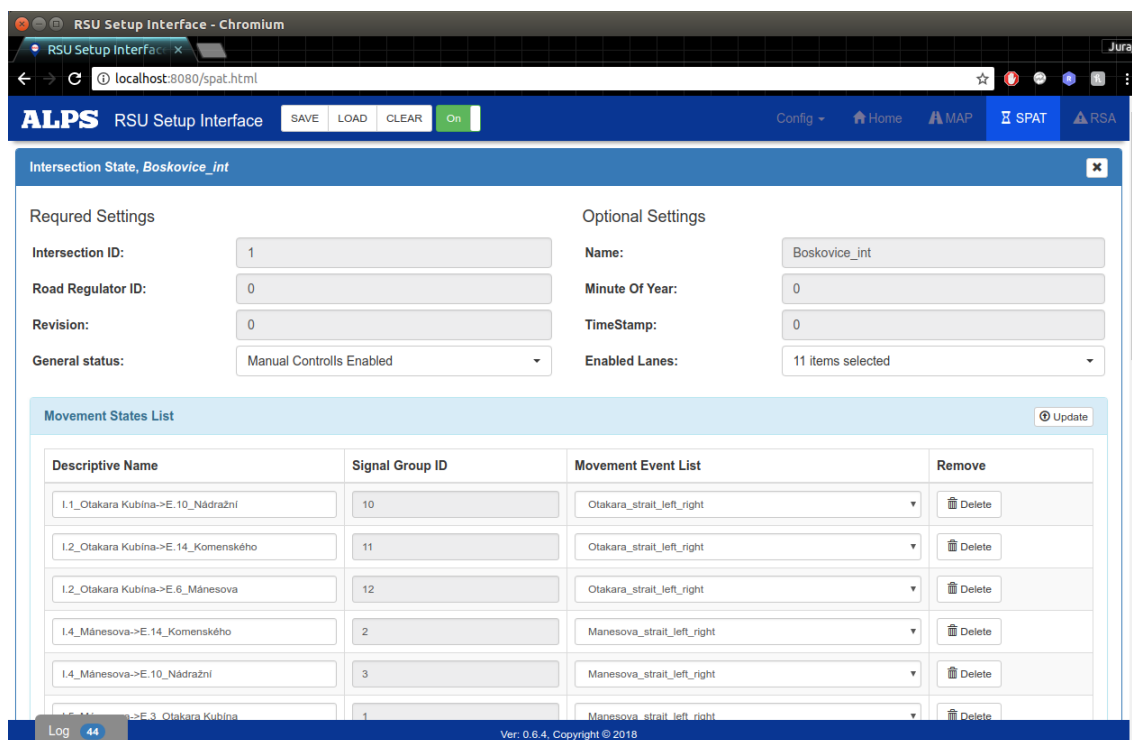
B UŽÍVATELSKÉ ROZHRAŇIE



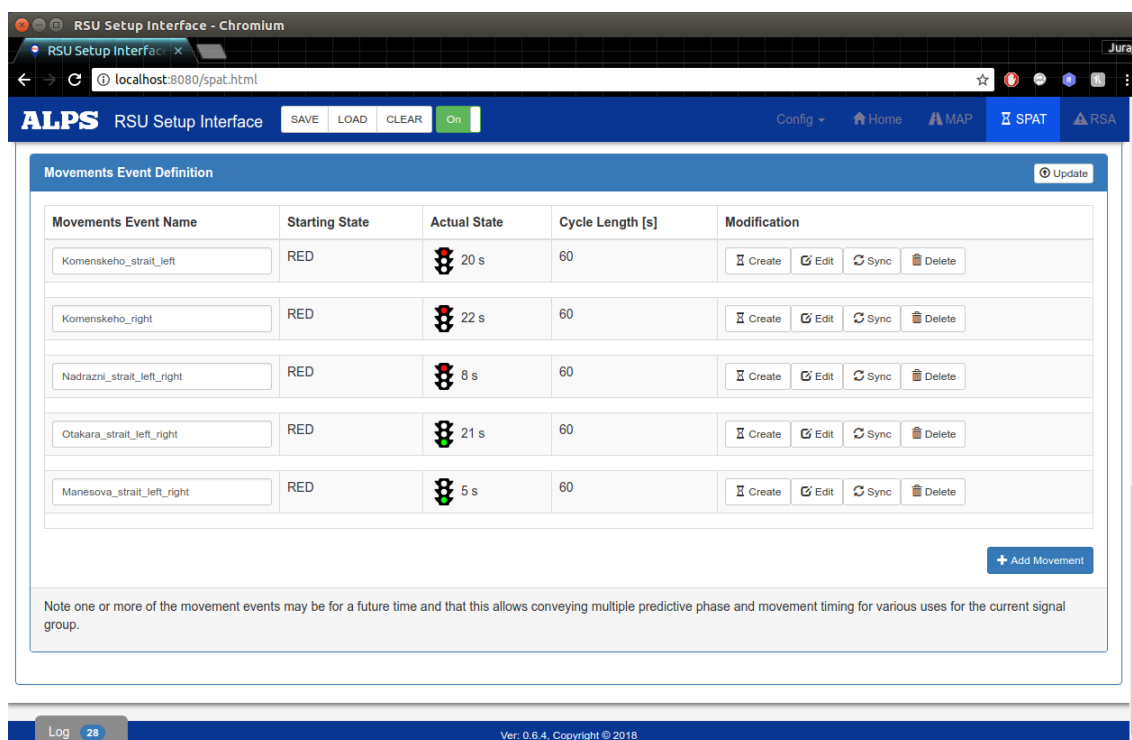
Obr. B.1: Grafické rozhranie pre nastavenie scenára.



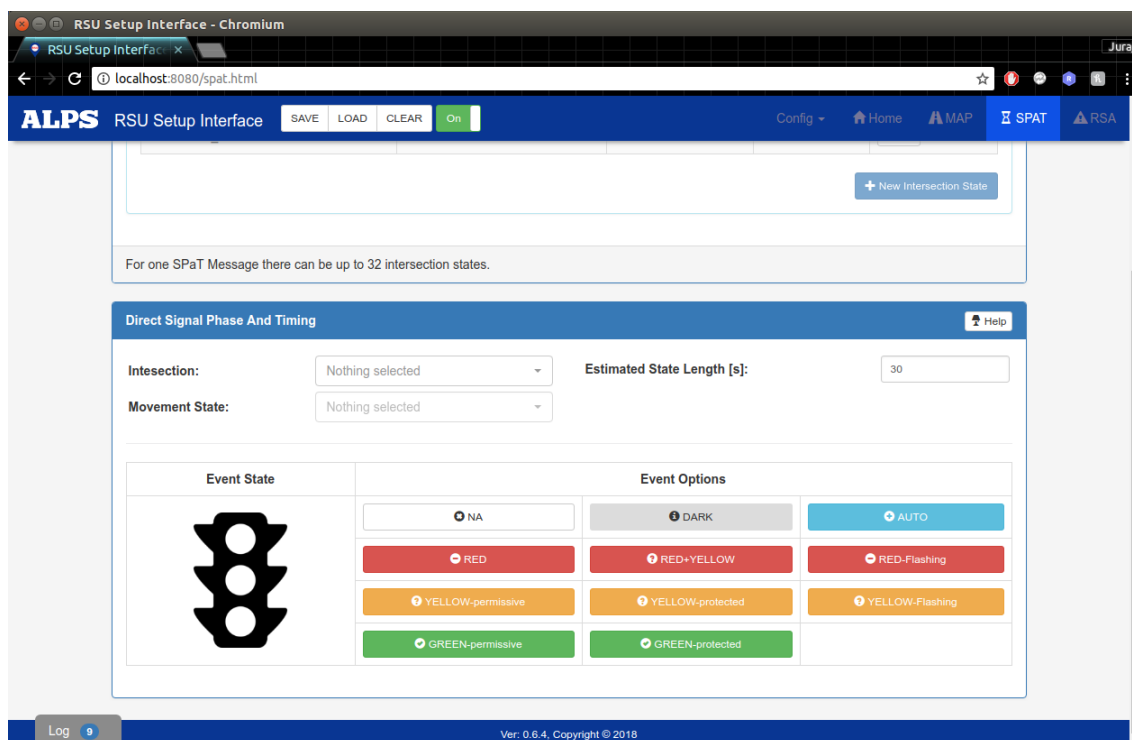
Obr. B.2: Grafické rozhranie pre definovanie MAP správach.



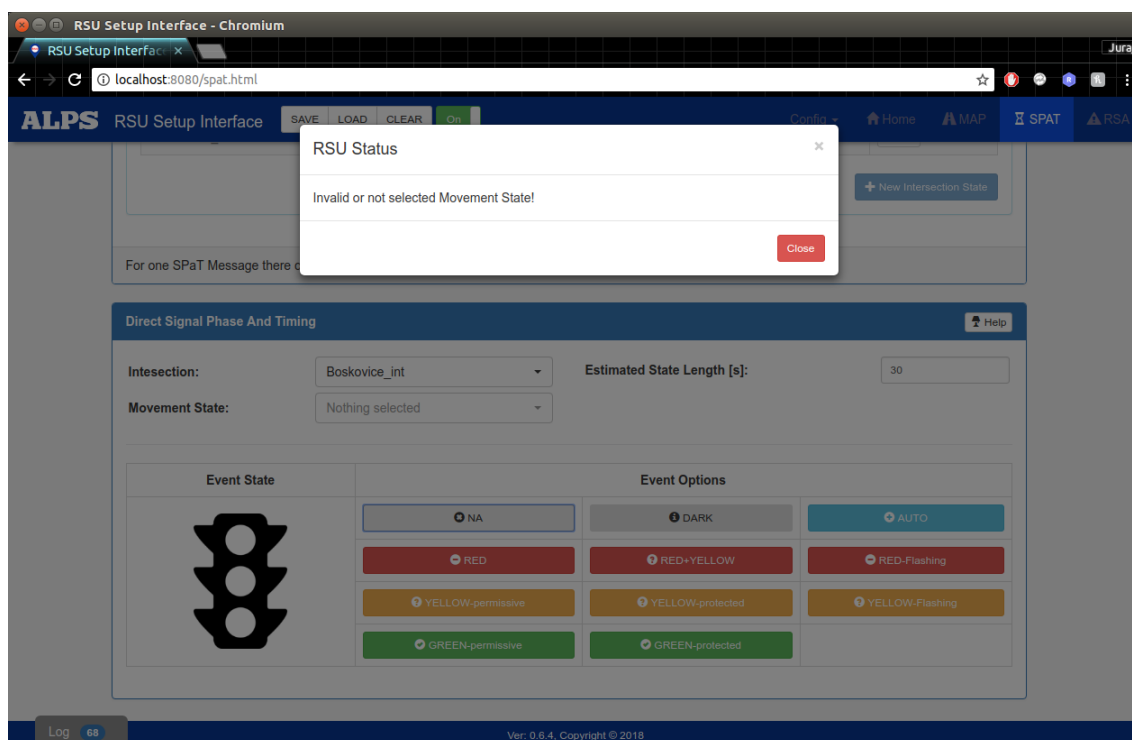
Obr. B.3: Grafické rozhranie pre priradenie udalosti k prepojeniam.



Obr. B.4: Grafické rozhranie pre špecifikovanie udalostí v SPAT správach.



Obr. B.5: Grafické rozhranie pre priamu modifikáciu udalostí v SPAT správach.



Obr. B.6: Upozornenie na nesprávne nastavenie.

C PRÍKLADY KONFIGURÁCIÍ

Výpis C.1: Príklad konfigurácie pre priame nastavenie stavu pohybu.

```
{
  "signalGroupID": 2,
  "intersectionID": 1,
  "auto": False,
  "movementEventsID": 2,
  "event": {
    "minEndTime": 10,
    "eventState": 3
  }
}
```

Výpis C.2: Základná štruktúra konfiguračného súboru.

```
"global": {
  "version": "0.6.5",
  "selectedScenario": 1,
  "availableScenarios": [ ... ]
},
"map": {
  "version": "0.6.3",
  "optional": {
    "timeStamp": 527040,
    ...
    "restrictionList": [ ... ]
  },
  "intersections": [ ... ]
},
"spat": {
  "version": "0.6.3",
  "name": "TopSPATMessage",
  "timeStamp": 527040,
  "intStateList": [ ... ]
}
"rsa": {
}
```

Výpis C.3: Štruktúra konfigurácie pre RSA modul.

```
"rsa": {
  "mapOpt": {
    "center": {
      "lat": 49.26477309971019,
      "lng": 16.56648963689804
    },
    "zoom": 13
  },
  "trigered": [],
  "rsa_messages": [
    {
      "required": {
        "typeEvent": 3
      },
      "optional": {
        "timeStamp": 1524906239131,
        "description": 3,
        "priority": 0,
        "heading": 255,
        "extent": 255,
        "position": {
          "lat": 49.26892467013629,
          "lng": 16.575571596622467
        },
        "icon": "/pic/den_pic/roadworks.ico"
      }
    }
  ]
}
```

Výpis C.4: Skrátená štruktúra konfigurácie pre SPAT modul.

```

"spat": {
  "version": "0.6.3",
  "name": "TopSPATMessage",
  "timeStamp": 527040,
  "intStateList": [
    {
      "required": {
        "intersectionID": 1,
        "revision": 0,
        "status": 1,
        "movementStatesMap": {},
        "movementStates": [
          {
            "required": {
              "signalGroupID": 1,
              "movementEventsID": 2
            },
            "optional": {
              "movementName": "I.1_Otakara Kubina->E.10_Nadrazni"
            }
          },
          // .
          // .
          // .
          {
            "required": {
              "signalGroupID": 10,
              "movementEventsID": 5
            },
            "optional": {
              "movementName": "I.4_Manesova->E.14_Komenskeho"
            }
          }
        ]
      },
      "optional": {
        "RoadRegulatorID": 0,
        "name": "Boskovice_int",
        "moy": 0,
        "timeStamp": 0,
        "enabledLanes": [
          2,
          4,
          5,
          11
        ],
        "maneuverAssistList": []
      },
      "movEventDefinitions": {
        "MovementEvents": [
          {
            "ID": 1,
            "name": "Komenskeho_strait_left",
            "timeStamp": 15240552659,
            "events": [
              {
                "eventState": 3,
                "minEndTime": 37
              },
              {
                "eventState": 4,
                "minEndTime": 2
              },
              {
                "eventState": 5,
                "minEndTime": 18
              },
              {
                "eventState": 7,
                "minEndTime": 3
              }
            ]
          },
          // .
          // .
          // .
          {
            "ID": 5,

```

```
"name": "Nadrazni_strait_left_right",
"timeStamp": 15240553115,
"events": [
  {
    "eventState": 3,
    "minEndTime": 38
  },
  {
    "eventState": 4,
    "minEndTime": 2
  },
  {
    "eventState": 5,
    "minEndTime": 17
  },
  {
    "eventState": 7,
    "minEndTime": 3
  }
]
}
```


D OBSAH PRILOŽENÉHO CD

Príloha je tiež dostupná na http://bit.ly/Priloha_Emulace_RSU.

/	Koreňový adresár priloženého CD
└─ obrázky/	Obrázky použité v texte
└─ Dokumenty/	
└─ v2x_rsu_docs.pdf	Dokumentácia zdrojového kódu aplikácie
└─ Softvér/	
└─ Docker/	Súbory pre preklad kontajnerov
└─ V2X-RSU.zip	Vyvinutá aplikácia (Python 3.5)
└─ testy/	
└─ TESTFEST_2/	
└─ Boskovice_int/	
└─ test_static_data/	
└─ README.md	Popis k testom
└─ priklady_konfiguracii/	
└─ boskovice_int.json	
└─ TestFest_France_ints.json	
└─ KrPole_test_ints.json	
└─ Ubuntu_16_V2X_RSU/	
└─ Ubuntu_16_V2X_RSU.zip	VirtualBox v5.1
└─ README.md	Prihlasovacie údaje
└─ Emulácia_RSU_pre_ITS.pdf	Tento text